HP 2393A Graphics Terminal HP 2397A Color Graphics Terminal Reference Manual

HEWLETT PACKARD

Manual Part Number:
02397-90002

Preface

This manual contains information on the HP 2393A Monochrome Graphics Terminal and the HP 2397A Color Graphics Terminal. Five of the twelve sections in the manual contain information which applies only to the color terminal. This informaiton appears at the end of sections 2, 3, 4, 9, and 11. All other information in the manual applies to both the monochrome and color terminals.

FCC Statement

Warning:

The US Federal Communications Commission (in 47 CFR 15.818) has specified that the following notice be brought to the attention of the users of this product.

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits of a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever steps may be required to correct the interference.

For Europe:

Verband Deutscher Elektroniker (VDE)

This equipment has been certified to comply with the German standard VDE level 0871B.

Notice

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing or use of this material.

Precision Visuals, DI-3000 and GRAFMAKER are trademarks of Precision Visuals, Inc.

ISSCO, DISSPLA and TELL-A-GRAF are registered trademarks of Integrated Software Systems Corporation.

SAS/GRAPH is a trademark of SAS Institute, Inc.

SPSS Graphics is a trademark of SPSS Inc.

NOVA*GKS is a trademark of Nova Graphics International Corporation.

TEKTRONIX is a registered trademark of Tektronix, Inc.

Roseville Terminals Division
8020 Foothills Blvd.
Roseville, CA 95816

Printing History

First Edition—September 1985
Printed in U.S.A.

02397-90002

Table of Contents

Section 1

Introduction

Section 2

Introduction to Graphics

Section 3

Graphics Operations

Section 4

Section 5

Terminal Control

Section 6

Configuring the Terminal

Section 7

External Devices

Section 8

Data Communications

Section 9

Status

Section 10

Block Data Transfers

Section 11

User-Definable Function Keys

Section 12

Designing and Using Forms

Appendix E

List of Illustrations

1 Introduction

Introduction

The 2393A terminal is a monochrome graphics terminal, consisting of a terminal processing unit (TPU), a monitor, and a keyboard. In addition to the keyboard, Hewlett-Packard Human Interface Link (HP-HIL) devices can be used to supply input. These devices include a touchscreen, graphics tablet, mouse, and bar code reader. The terminal can supply alphanumeric output to a printer and graphics output to a printer or plotter.

This section describes the terminal features, input and output devices, the primary terminal modes, terminal memory and the terminal specifications.

Graphics Features

The terminal graphics features are as follows:

- Raster display with selectable resolution: 512H X 390V and 640H X 400V.
- Rubberband line graphics, giving visual feedback for line drawing before a line is stored in memory.
- Eleven line types for line drawing, including seven predefined line types, a solid line, a user defined line type, the current area fill pattern used as a line type, and a point.
- Polygonal area filling using one of eight predefined area fill patterns, a user defined pattern, or a solid area fill.
- Graphics vector text, with variable character orientation and size. Supports all national languages supported by the terminal.

- Software support for HP DSG/3000, HP Graphics 1000/II, HPDRAW, HPEASYCHART, and HPMAP.
- Tektronix 4010/4014 Compatibility mode, which extends the number of addressable screen points to either 1024 X 1024 (4010 emulation) or 4096 X 4096 (4014 emulation).

Alphanumeric Features

The terminal alphanumeric features are:

- Screen capacity of 24 lines of 80 characters, with three additional lines for function key labels and a terminal status line.
- Selectable 80 to 160-character wide workspace with horizontal scrolling.
- Display control, including smooth scrolling, next vertical page, previous vertical page, next horizontal page (left or right), roll up and roll down, and clear display
- Display enhancements, including inverse video, half-bright, security video, blinking, underline, and double-high double-wide characters.
- Line drawing, math, italics, and bold character sets.
- Programmatic cursor sensing and addressing.
- Character mode or Block mode (Line or Page) operation.
- Editing operations, including character insert and delete (with and without wrap) and line insert and delete.
- Format mode, with edit checks, security fields, and modified data tags, for data entry.
- Optional ANSI X3.64 and VT52 compatibility.
- Bar code reader input capability.

Terminal Features

- Independent graphics and alphanumeric display memories.
- USASCII or any of 16 optional national keyboards.
- Twelve user definable function keys. Labels for f1–f8 displayed on the screen. Four keys (f9–f12) executable, automatically, at power on.
- User definable (with two characters) Return key.
- Integrated terminal family (ITF) standard keyboard with keypad for numeric data entry, graphics control, or (optional) ANSI applications.
- All configuration done on screen displayed menus. Configuration data stored in nonvolatile memory.
- Extensive self-test capability.

Options

Terminal options are as follows:

OPTION	FEATURE
Std	Monochrome graphics terminal Keyboard: USASCII. Port 1: RS232C or RS422 datacomm port.
049	ANSI X3.64 Operation.
060	Delete monitor.
046	Port 2: HP-IB Interface Module.
092	Port 2: 25-pin RS232C Interface Module.
093	Port 2: 8-bit parallel Centronics Interface Module.
101	Swedish
102	Norwegian
103	French
104	German
105	United Kingdom
106	European Spanish
107	French Canadian
108	English Canadian
109	Italian
110	Dutch
111	Finnish
112	Danish
113	German Swiss
114	French Swiss
115	Latin American Spanish
116	Flemish
301	US modem cable.
302	European modem cable.
303	RS232C cable.
304	HP direct-connect RS232C cable.
305	EMP protect cable.
306	HP direct-connect RS422 cable.

Input Devices

In addition to the keyboard, the terminal accepts input from several Human Interface Link (HP-HIL) devices. These are:

- Touchscreen 35723A.
- Bar code reader 92916A.
- Tablet 46087A or 46088A.
- Mouse 46060A.

Several input devices, including the keyboard, can be connected to the terminal at a time.

When equipped with a touchscreen, the terminal can operate in two modes: Alphanumeric or Touch Mouse. In Alphanumeric mode, the terminal is set up to operate on alphanumeric data. Touch Mouse mode is for graphics data entry.

In Alphanumeric mode, the terminal has several capabilities:

- Touch-sensitive fields can be programmatically defined on the screen so that, when touched, the terminal can be made to send a string of characters to a program operating on the host computer. These characters can be used by the program to initiate any action of which the terminal is capable, such as displaying a data entry form, etc.
- All unprotected fields on a data entry form can be made touch sensitive, for cursor positioning by the host computer program.
- Function keys f1–f8 can be made touch sensitive, so that the operation assigned to them, whether system or user defined, will be triggered by a touch.

In Touch Mouse mode, a touch on the screen will move the graphics cursor to the touched point on the screen. A program operating on the computer can sense the current cursor position. By sensing the cursor position at small time

intervals, a host computer program can enable the user to draw lines on the screen with a finger, as though the input came from a tablet or mouse.

The bar code reader enables quick and easy alphanumeric data entry.

The tablet and mouse are both graphics input devices, which send coordinates to the computer. As in Touch Mouse mode, these coordinates can be used by a program operating on the host computer to position the graphics cursor on the screen and draw lines.

Refer to Appendix E for details on input device use.

External Devices

Printers can be used to copy both alphanumeric and graphics data from the terminal. Plotters can be used to copy graphics data, provided they are supported by a host computer program, such as HPDRAW, to perform the necessary interface operations between the terminal and plotter.

External devices are normally connected to port 2, which accepts one of three optional interface modules. Options 046 is for connecting printers or plotters in an HP-IB network. With option 092, an RS232C interface, installed in the terminal, either port can be connected to the computer, with the other port connected to an RS232 printer or plotter. Option 093 is for connection of 8-bit parallel input Centronics-type printers.

Configuration

Configuration of the terminal is performed using menus. These include a Global Configuration menu, a Terminal Configuration menu, a hardwired data communications (datacomm) menu, a modem datacomm menu, three external device configuration menus, and an ANSI configuration menu. Only one of the three external device menus can be displayed, depending on the interface module currently installed in the terminal:

OPTION	MENU
046	External HPIB Device Configuration
092	External Serial Device Configuration
093	External Parallel Device Configuration

Terminal Modes

The terminal has three primary operating modes, HP, ANSI, and EM52. ANSI mode allows use of the terminal with a computer system which uses ANSI protocol. In EM52 mode, the terminal emulates a DEC VT52®-compatible terminal. Appendix B contains details on ANSI and EM52 mode operation. In HP mode, the terminal operates like a terminal without the ANSI option. All information in this manual, except for Appendix B, applies to HP mode.

The standard terminal (in HP mode) can operate in Graphics mode, Alphanumeric mode, or Compatibility mode. Graphics mode is for graphics operations, Alphanumeric mode is for alphanumeric data operations, and, in Compatibility mode, the terminal can be used with programs written for Tektronix 4010 and 4014 terminals.

The primary alphanumeric modes are listed below:

- Local — Data entered from the keyboard is displayed on the screen, but the terminal is disconnected from the host computer.
- Remote — The terminal is connected to the host computer. Data entered from the keyboard is transmitted to the computer, and data received from the computer is displayed on the screen.
- Character — Active only in Remote mode. Data is transmitted to the host computer, one character at a time.
- Block — Active only in Remote mode. Data is transmitted to the computer in blocks; selectable as one line per block, or as one page per block.
- Line — Active only in Remote Block mode. Data is transmitted to the host computer in blocks, each consisting of one line.
- Page — Active only in Remote Block mode. Data is transmitted to the host computer as a block, consisting of one page (one screenful).
- Format — Used to control input and formatting of data entered into data entry forms, which are displayed on the screen. Enables restriction of entry data to selected fields on the form, and restriction to selected data types (numeric, alphabetic, or unrestricted).
- Non-Format — Terminal operates normally. No restriction on data entry.

Memory

Terminal memory consists of the following types:

ROM —	Contains the terminal operating code.
SYSTEM RAM —	Scratch memory for internal firmware use (storing variables, etc.)
Graphics —	Contains graphics data for display. The data can be entered using an input device or from a program. The capacity of graphics memory is one screenful of data.
Alphanumeric —	Contains alphanumeric data for display on the screen. The data can be entered from the keyboard, or other input device, or from a program. Its capacity is somewhat variable, depending on the amount of display enhancements used. Another name for alphanumeric memory is display memory.
Nonvolatile —	A portion of memory used to retain stored data during power-down periods. It is used to store configuration selections.

Escape Sequences

Escape sequences, incorporated into a computer program, enable the program to control terminal operations. When the terminal receives the escape sequence, it performs the operation specified in the sequence. Escape sequence operations consist of most of the operations performable at the keyboard, using the non-alphanumeric keys. Many sequences are also enterable from the keyboard. References to escape sequences are made through out the manual. Refer to Appendix A for a complete list of the escape sequences, their functions, and rules for use.

Specifications

Terminal specifications are listed in table 1-1.

Table 1-1. Specifications

GRAPHICS:

Viewable Resolution:	512(H) x 390(V) pixels, or 640(H) x 400(V) pixels, selectable.
Addressability:	32K x 32K points in HP mode.
	4096 x 4096 points in Compatibility 4014 mode.
	1024 x 1024 points in Compatibility 4010 mode.
Display Area:	160mm x 214mm (6.3 inches x 8.4 inches) or 131mm x 201mm (5.2 inches x 7.9 inches), selectable.
Cursor:	Crosshair.
Primitives:	
Polyline:	11 predefined line types. User defineable lines.
Area fill:	Fills polygons with up to 148 sides using either eight predefined patterns or user defined area fill paterns.
Text:	Eight sizes, normal or slant, four orientations, 10 fully supported national character sets.

ALPHANUMERICS:

Display Area:	155mm x 201 mm (6.1 inches x 7.9 inches) or 124mm x 201mm (4.9 inches x 7.9 inches), selectable.
Character Generation:	7 x 11 (includes descenders) dot matrix in 8 x 14 dot cell with half-dot shift.
Character Size:	1.9 mm x 3.7 mm (0.075 inch x 0.146 inch.
Screen Capacity:	24 lines x 80 columns (1920 characters). 25th and 26th lines for function key labels. 27th line for terminal status information.

Table 1-1. Specifications (continued)

Character Set:	ROMAN 8 comprising 128 ASCII character set and 61 national characters from the extended Roman set. National characters include: English, Italian, Danish, Dutch, German, Norwegian, Spanish, Finnish, French, Swedish
Cursor:	Blinking-underline or blinking-block.
Display Enhancements:	Inverse, underline, blinking, halfbright, security video, and double-high double-wide in any combination (double-high double-wide applies only to whole lines).
Video Attributes:	Memory lock, display lock, protected fields, user selectable margins and tabs, plus jump or smooth scrolling.
Editing:	Insert character/line, delete character/line, clear line/all, insert/delete character with wraparound.
Page Control:	Next/previous page (top, bottom, right, left).
Display Memory:	Memory is allocated dynamically. Up to 12 pages standard (288 lines x 80 columns or 144 lines x 160 columns). Depending on the number of display enhancements and unprotected fields used, memory available can vary.

DISPLAY:

CRT Type:	Etched anti-glare, with implosion protection.
Phosphor:	Green P31.
CRT Size:	30 cm (12 inches) diagonal.
Refresh Rate:	60 Hz non-interlaced.

Table 1-1. Specifications (continued)

DATA COMMUNICATIONS:	
Data Rates:	System port: 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200 baud. Operation with control codes or escape sequences may require handshaking or CPU-supplied delays. Optional peripheral port with RS232C interface: 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800,9600, 19200 baud. Operation with control codes or escape sequences may require handshaking or CPU-supplied delays.
Handshake:	System port: selectable software ENQ/ACK and XON/XOFF (transmit only), hardware CS (Clear to Send), TR (Terminal Ready), DM (Data Set Ready), and RR (Receiver Ready). Optional peripheral port: selectable software XON/XOFF (transmit only), hardware CS (Clear to Send), SRR (Secondary Receiver Ready), and DM (Data Set Ready).
Parity:	Selectable in 7-bit operation: even, odd, zero, one. Always none in 8-bit operation.
Operating Modes:	Local, Remote, Character, Block Line or Page, Format, and optional ANSI operation. X.25-compatible using XON/XOFF when Block mode is enabled and when the terminal is connected to an external PAD.
Transmission Modes:	Full duplex, asynchronous point-to-point. Compatible with Bell 103A, Bell 212A, Vadic VA3400 or equivalent full duplex modem.
Electrical Interface:	System port: combined RS232C/HP422 data communications. Optional peripheral port: RS232C serial interface, Centronics-type parallel 8-bit interface, or HP-IB interface.
POWER REQUIREMENTS:	
Input Voltage:	100–120V (+10%, −10%) at 47–66 Hz, or 200–240V (+10%, −10%) at 47–66 Hz, switch selectable.
Power:	50 watts, Terminal Processing Unit 45 watts, Display Monitor.

Table 1-1. Specifications (continued)

PHYSICAL CHARACTERISTICS:

Display Monitor:

Weight:	10.0 kg (22 pounds).
Dimensions:	328 mm wide x 351 mm deep x 345 mm high (12.9 inches x 13.8 inches x 13.6 inches).

Terminal Processing Unit:

Weight:	4.3 Kg (9.5 pounds).
Dimensions:	325 mm wide x 325 mm deep x 100 mm high (12.8 inches x 12.8 inches x 3.9 inches).

Keyboard:

Weight:	2.1 kg (4.7 pounds).
Height (at center row):	30 mm (1.2 inches) flat; 58 mm (2.3 inches) on stand.
Dimensions:	456 mm wide x 225 mm deep x 30 mm high (18.0 inches x 8.9 inches x 1.2 inches).

ENVIRONMENTAL CONDITIONS:

Altitude:

Operating:	0 to 4,600 meters (15,000 ft).
Non-operating:	0 to 15,300 meters (50,000 ft).

Humidity:

Operating:	5 to 95% Relative Humidity at 40 degrees C.
Non-operating:	90% Relative Humidity at 65 degrees C.

Temperature (free ambient):

Operating:	0 degrees C to +55 degrees C (+32 degrees F to +131 degrees F).
Non-operating:	−40 degrees C to +75 dgrees C (−40 degrees F to +167 degrees F).

2 Introduction to Graphics

Introduction

This section supplies an overview of terminal graphics and describes some of the fundamental operations used for graphics control, such as:

- How to communicate with terminal graphics from the keyboard and from a program.
- Control of the graphics display.
- Control of the graphics cursor.
- Drawing modes.
- Notation for points, used in drawing lines.

Overview

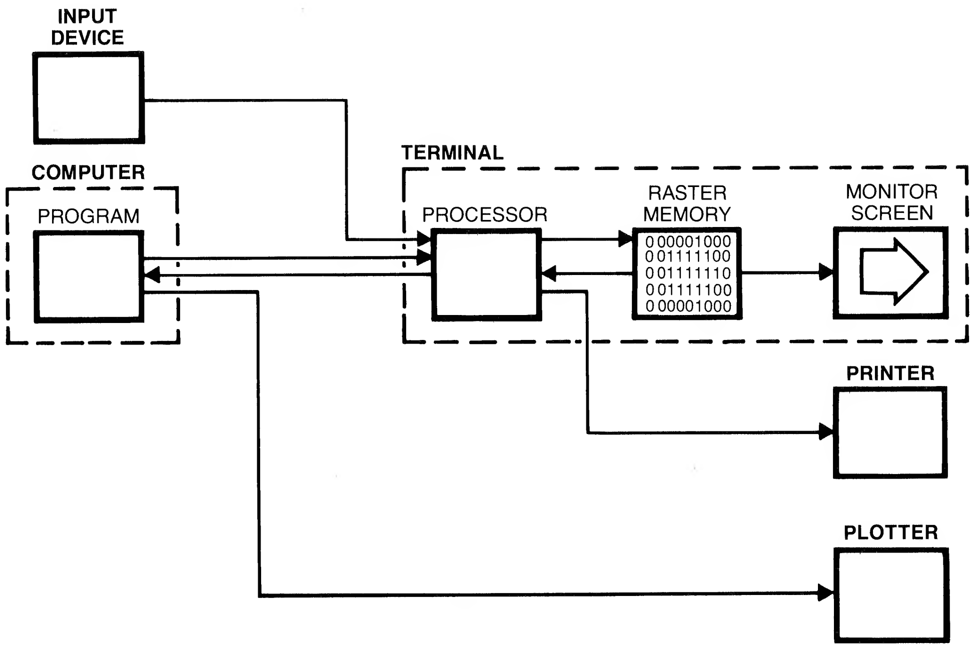
The following paragraphs provide a brief sketch of the terminals graphics capabilities and how they are used.

Terminal Input, Storage, and Output

A program operating on a host computer is the primary source of control and data for the terminal's graphics system. However, most of the programmatic instructions can also be generated from the keyboard. A graphics tablet, mouse, or touchscreen can also be used to position the cursor and draw lines.

A portion of the terminal's graphics memory, called raster memory, is used to store the picture data to be displayed on the screen. Raster memory consists of an array of bits, each bit corresponding to one pixel on the display screen. As a picture is created, the data from the input device is stored in raster memory (figure 2-1).

Figure 2-1. Graphics Data Input, Storage, and Output



When graphics memory is selected for display, raster memory is constantly scanned and it's contents displayed on the screen. For 1 bits, the associated pixel is lighted; for 0 bits, it is unlighted. A blank display results when raster memory is filled with 0's.

The contents of raster memory can be copied to a suitable printer.

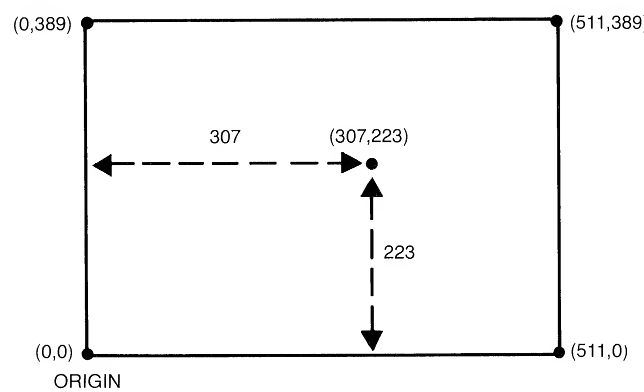
The Display Screen

Raster memory size is limited to the amount required to display one full screen of data.

Terminal screen resolution is selectable as either 512 (horizontal) by 390 (vertical) or 640 by 400 pixels.

The screen origin is at the lower left corner of the screen (figure 2-2). A point on the screen is specified in terms of it's distances from the origin in the horizontal (X) and vertical (Y) dimensions. These distances are called coordinates. By convention, the X coordinate is always listed first.

Figure 2-2. Screen Origin and Example Coordinates (Low Resolution Shown)



Picture Creation

Pictures are created by drawing lines, filling in areas, and (optionally) adding text.

Line Drawing. Lines can be drawn using any of the input sources: a program, the keyboard, a tablet, a mouse, or the touchscreen. Isolated points are treated as a type of line.

The terminal uses the concept of a pen for drawing lines. The pen has two positions: up and down. Lines are drawn by moving the pen in the down position. When the pen is moved, it is always left in the down position, ready to draw a line. One escape sequence can be used to draw a series of connected lines (polyline) by specifying, in a series, the end points of the lines. (The start point of each line is the end point of the last one.)

Area Filling. Areas are filled by two methods: rectangular area fills and polygonal area fills. Rectangular areas are filled by specifying two corners of the rectangle. Polygons are filled by specifying the end point of each straight line used in forming the external boundary of the polygon.

The pattern with which the area is filled must be selected before the area is specified. The terminal always uses the currently selected area fill pattern to fill the area. The pattern can be a solid area fill, one of eight predefined area fill patterns or a pattern you have defined yourself.

Adding Text. Text can be entered into graphics memory, from either the keyboard or a program, to be used as labels, titles, notes, etc. in the picture. Useable characters are upper and lower case characters from either the ASCII character set or any of the national character sets selectable on the Terminal Configuration menu. (Refer to Section 6, Configuring the Terminal, for the available character sets.)

The text can be used in eight sizes. It can be slanted 27 degrees, for an italics-like effect, or the line of text can be rotated, in 90-degree increments.

Relocatable Origin

Lines are drawn, programmatically, by specifying the X and Y coordinates of the end point of each line. A symbol or picture fragment, such as a company logo, can be contained in a program in terms of line end point coordinates.

Sometimes, it can be useful to position such a picture fragment at several locations on the screen. Terminal graphics contains a feature, called a relocatable origin, which makes this operation easy.

As previously discussed, the absolute (default) origin to which screen coordinates are referenced is the lower left corner of the screen. If a program containing a picture fragment is run, the picture fragment will be located as specified by the line coordinates contained in the program.

However, by specifying a relocatable origin in terms of X and Y offsets from the absolute origin, the picture fragment can be relocated on the screen. The new location is determined by adding the X coordinate of the relocatable origin to all X coordinates specified in the program for the picture fragment. All Y coordinates are handled the same way.

Obtaining a Hardcopy

The contents of raster memory, as displayed on the screen, can be reproduced by a raster-dump type printer connected to the terminal. Printer copy can be produced either from the keyboard or by sending the terminal a print command, in the form of an escape sequence, from a program operating on a host computer. Plotter copy can be produced by running a program on a host computer, such as HPDRAW, which sends commands to the plotter which duplicate what has been produced in raster memory.

Obtaining Graphics Status Information

Using an escape sequence, a program can obtain information such as the terminal graphics capabilities, the current location of the graphics cursor and pen, whether the pen is up or down, whether the terminal is set for high or low resolution, the currently selected graphics text options (size, slant, etc.), the location of the relocatable origin, and if a graphics positioning device is connected to the terminal.

Compatibility Mode

The terminal has a mode, Compatibility mode, which enables running programs written for a Tektronix 4010 or 4014 terminal on this terminal with minimum program alteration. This includes 4014 terminals with the Enhanced Graphics Module, which increases the 4014 resolution from 1024 X 1024 to 4096 X 4096.

Communicating with Graphics

You communicate with the terminal graphics firmware either through the keyboard or from a program.

From the Keyboard

All of the graphic function commands can be entered from the terminal keyboard by the operator using either escape codes, the set of graphics control keys located on the keyboard to the right of the ASCII character set, or selected alphanumeric keys. Table 2-1 contains a list of the keys and a description of their functions. Keys followed by an asterisk are graphics keypad keys; those without an asterisk are alphanumeric keys.

Table 2-1. Graphics Control Keys

KEY	DESCRIPTION
[Graph cursor]* [CTRL]/[Next]	Toggles the graphics cursor on and off.
[▲]* [CTRL]/[▲]	Move graphics cursor up.
[◄]* [CTRL]/[◄]	Move graphics cursor left.
[►]* [CTRL]/[►]	Move graphics cursor right.
[▼]* [CTRL]/[▼]	Move graphics cursor down.
[Cursor fast]*	Speeds up the graphics cursor if pressed in conjunction with the cursor movement keys. The rate returns to normal when released.

[Graph display]* [CTRL]/[Delete char]	Toggles the graphics display on and off without erasing it from memory.
[Alpha display]* [CTRL]/[Insert char]	Toggles the alphanumeric display on and off.
[Graph clear]* [CTRL]/[Clear display]	Erases graphics memory.
[Graph copy]* [CTRL]/[Enter]	Copies graphics memory to the selected "to" device.
[Move]*	Moves pen to current cursor
[Draw]*	Lowers pen and draws a line from the current pen position to current cursor position.
[RB Line]*	Toggles display of the rubberband line and the graphics cursor on and off. The rubberband line is a line between the current pen and cursor positions.
[Graph device]*	Toggles the connection between any HP-HIL graphics input devices and the terminal on and off.
[CTRL]/[◸]	Graphics home up. Moves the graphics cursor to the top left corner of the screen.
[CTRL]/[Shift]/[◸]	Graphics home down. Moves the graphics cursor to the bottom left corner of the screen.
[Shift]/[NUM -] or [CTRL]/[NUM -]	Toggles the function of the keypad between graphics and numerics.
	When unshifted in numeric mode, the key is used to display a dash (-) character.

The graphics keypad functions also as a numeric keypad. The keys [Shift] [NUM] or [CTRL] [NUM], pressed together, toggle the keypad functions between graphics and numerics. The current mode, alphanumeric or graphics, is saved in non-volatile memory and returned at power on or after a hard reset.

When the keypad is in graphics mode, the minus function on the [NUM] key, and the [7], [8], and [9] keys are disabled.

From a Program

Graphics functions are controlled by escape sequences, which all begin with `Ec *`. The third character, always lower case, selects the type of graphics sequence. Table 2-2 lists the types of graphics sequences. For example, `Ec*p` specifies a plotting sequence.

Table 2-2. Summary of Graphics Sequence Types

ESCAPE SEQUENCE	TYPE
Ec*d	Display Control
Ec*j	Input Device Control
Ec*l	Labeling
Ec*m	Drawing Mode
Ec*p	Line Plotting
Ec*s	Graphics Status
Ec*t	Compatibility Mode

Subsequent characters in the control sequence are read as either parameters or commands, depending on the location of the character in the ASCII table (figure 2-3).

Figure 2-3. ASCII Characters Used in Escape Sequences

BIT 765 / 4321	000	001	010	011	100	101	110	111
0000	NUL NU	DLE DL	SP	0	@	P	`	p
0001	SOH SH	DC1 D1	!	1	A	Q	a	q
0010	STX SX	DC2 D2	"	2	B	R	b	r
0011	ETX EX	DC3 D3	#	3	C	S	c	s
0100	EOT ET	DC4 D4	$	4	D	T	d	t
0101	ENQ EQ	NAK NK	%	5	E	U	e	u
0110	ACK AK	SYN SY	&	6	F	V	f	v
0111	BEL	ETS ES	'	7	G	W	g	w
1000	BS BS	CAN CN	(	8	H	X	h	x
1001	HT HT	EM EM	)	9	I	Y	i	y
1010	LF LF	SUB SB	*	:	J	Z	j	z
1011	VT VT	ESC EC	+	;	K	[	k	{
1100	FF FF	FS FS	,	<	L	\	l	¦
1101	CR CR	GS GS	–	=	M	]	m	}
1110	SO SO	RS RS	.	>	N	∧	n	~
1111	SI SI	US US	/	?	O	—	o	DEL

Parameters (columns 010, 011) — Commands (columns 100–111)

BIT 7 6 5 4 3 2 1

Bits 7 6	Meaning
0 0	Control Code
0 1	Parameter
1 0	Command and Terminate Sequence
1 1	Command and Continue Sequence

Control Codes. Control codes are generally ignored, with the exception of the ESCAPE character (Ec). If an Ec character is detected and the previous graphics control sequence has not been properly terminated with a "Z" or some other valid upper case character, the Ec character will cause the execution of the previous sequence to be terminated. The new escape sequence will then be executed.

Commands. Graphics commands come from columns 4-7 of the ASCII table, the upper and lower case letter (A-Z and ^). Both upper and lower case commands execute the same function. Upper case letters terminate the sequence and cause it to be executed. You can use more than one command in a sequence.

Graphics sequences can be any length. (The terminal ignores CR and LF characters in the middle of graphics sequences.) For example, to plot a figure containing 100 points the escape sequence could appear as follows:

```
Ec*p a <x1,y1> . . . <x100,y100>Z
```

This could cause problems if an error occurs and the system tries to report it in the middle of a long sequence. Since most systems use upper case characters for messages, the first character of the message would end any graphics sequence that might be in progress. Letters that have not been assigned a function for a particular graphics sequence are treated as NOP's and if they are lower case, are ignored. If upper case, they will end the sequence. The letter z has been defined as a NOP in all sequences so that a capital Z can always be used to end a graphics escape sequence.

Parameters. Parameters come from columns 2 and 3 of the ASCII table (`SPACE` through `?`). Most parameters are simply the ASCII numeric characters used to represent data coordinates or to select one of several settings. Binary formatted data is generated by appending the bits 0 1 to five bits of binary data. Note that in binary formats, all ASCII characters with decimal codes from 32 through 63 (`SPACE` through `?`) are treated as data and are not ignored or used as delimiters. Both ASCII and binary data formats are described later in this section.

Since a number of parameters grouped together in an escape sequence can be confusing and difficult to read, use of spaces or some other non-digit, non-letter ASCII character is recommended to separate parameters. These characters are ignored by the terminal.

Parameters precede their associated commands (postfix notation). The most frequently used parameters are point coordinates.

Examples:

```
              commands

   Ec*m  2a  7b   256,195  J
 preamble

                  parameters
```

```
   command               end sequence

Ec*p  f  100, 100, 200, 200Z
preamble         parameters
```

Graphics Display Control

The resolution of the graphics display can be increased, the graphics display and graphics cursor can be turned on or off, or the entire memory can be set to all ones (pixels on) or all zeros (pixels off).

Changing the Resolution of the Display

The resolution of the graphics display can be increased from the default value of 512 pixels (horizontal) and 390 pixels (vertical) to 640 (horizontal) and 400 (vertical). It can also be returned to the default values.

To change the resolution on monochrome terminals, a switch on the rear panel of the monitor (HP monitors) must be set to the appropriate position (high or low) and an escape sequence sent to the terminal. (This switch is not present on color terminals.) The escape sequence is as follows:

```
Ec*d <low x><low y><high x><high y> y
```

where:

`<low x>` = 0
`<low y>` = 0
`<high x>` = 511 or 639
`<high y>` = 389 or 399

Graphics Display On/Off

From the graphics/numeric keypad, pressing GRAPH DSPLY toggles the graphics display on and off. Data stored in graphics memory is not affected by this action.

Programmatically:

Graphics Display On:	`Ec*dC`
Graphics Display Off:	`Ec*dD`

Graphics Display Set/Clear

From the keyboard, you may clear graphics data on the screen, to produce a black screen, by pressing GRAPH CLEAR on the graphics/numeric pad.

To do the same thing, programmatically:

Clear Graphics Memory: `Ec*dA`

To set graphics memory, programmatically:

Set Graphics Memory: `Ec*dB`

Alphanumeric Display On/Off

You can turn the alphanumeric display on or off. Neither the graphics nor the alphanumeric data are affected by these operations. The escape sequences are as follows:

On	`Ec*dE`
Off	`Ec*dF`

Graphics Cursor Control

A graphics cursor, unrelated to the alphanumeric cursor, is available for use in locating points in the graphics display. You use it to input position data or to interact with a graphics application program.

Graphics Cursor On/Off

The graphics cursor is initially off (power on or full reset). Turning the cursor on or off does not affect the data in graphics memory.

The graphics cursor may be toggled on and off by pressing the [GRAPH CURSOR] key on the graphics/numeric pad.

Programmatically, you can toggle the cursor:

`Ec*dK`	Graphics Cursor On
`Ec*dL`	Graphics Cursor Off

Graphics Cursor Positioning

Several means are available for positioning the graphics cursor: the keyboard, a mouse, tablet, touchscreen, or a program. Refer to table 2-1 for information on keyboard cursor positioning and to the User manual for mouse, tablet, and touchscreen information.

The graphics cursor is initially at position (0,0) after power on or a full reset.

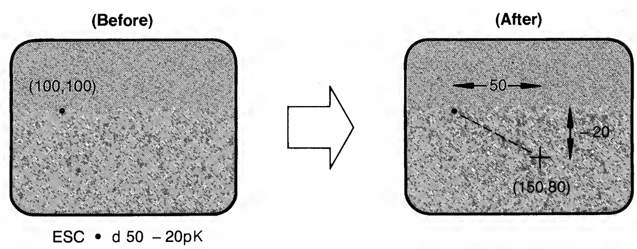
From the keyboard, the position of the graphics cursor is controlled using the [▲], [▼], [◄], and [►] keys on the graphic/numeric keypad. Pressing two keys simultaneously will cause diagonal cursor movement. The [CURSOR FAST] key may be pressed, simultaneously, to speed up cursor positioning. The cursor position, displayed at the bottom center of the screen, can be used as an aid in cursor positioning.

Programmatically, the cursor can be positioned (even if it is not turned on) using either absolute or relative coordinates. In the following sequences X and Y give the new cursor position.

Position Graphics Cursor Absolute:	`Ec*d<X,Y>O`
Position Graphics Cursor Relative:	`Ec*d<X,Y>P`

Example: The cursor is currently at position 100,100 and is off. Move it 50 units to the right and 20 units down from its current position and turn it on (figure 2-4).

Figure 2-4. Cursor Positioning Example



Alphanumeric Cursor On/Off

The alphanumeric cursor can be turned on and off in graphics mode.

`Ec*dQ`	Turn alphanumeric cursor on.
`Ec*dR`	Turn alphanumeric cursor off.

Drawing Modes

The terminal has five drawing modes. These modes determine the way data is displayed on the screen, so that changing the drawing mode can radically change the screen appearance of a given set of data. The modes are:

NO.	NAME
0	Picture Protect
1	Clear
2	Set
3	Complement
4	Jam

Several definitions are required for the following discussion:

- Entered data — Data entered into the terminal.
- Existing data — Data existing on the screen when the entered data is received.
- New data — Data displayed on the screen as a result of the entered data.

In modes 1 through 3, a 1 bit in the entered data is a signal to the terminal to set the corresponding existing data bit to a 1 or 0, depending on the mode. For 0 bits in the entered data,the existing data is not affected. For each 1 bit, the screen pixel is on; for each 0 bit, the pixel is off.

For mode 4, Jam mode, the data existing on the screen is converted to the entered data.

The default mode, which is selected after a power on or hard reset, is mode 2 (Set mode). In this mode, 1 bits in the entered data are drawn as 1's (pixel on); 0 bits have no effect.

The escape sequence for selecting the drawing mode is as follows:

```
Ec*m <x>a
```

where <x> is a number which selects the mode:

<x>	MODE
0	Picture Protect
1	Clear
2	Set
3	Complement
4	Jam

Figure 2-5 illustrates the effect of each mode on a given set of data. Table 2-3 supplies an example for each mode.

Figure 2-5. Effect of Each Drawing Mode on a Given Set of Data

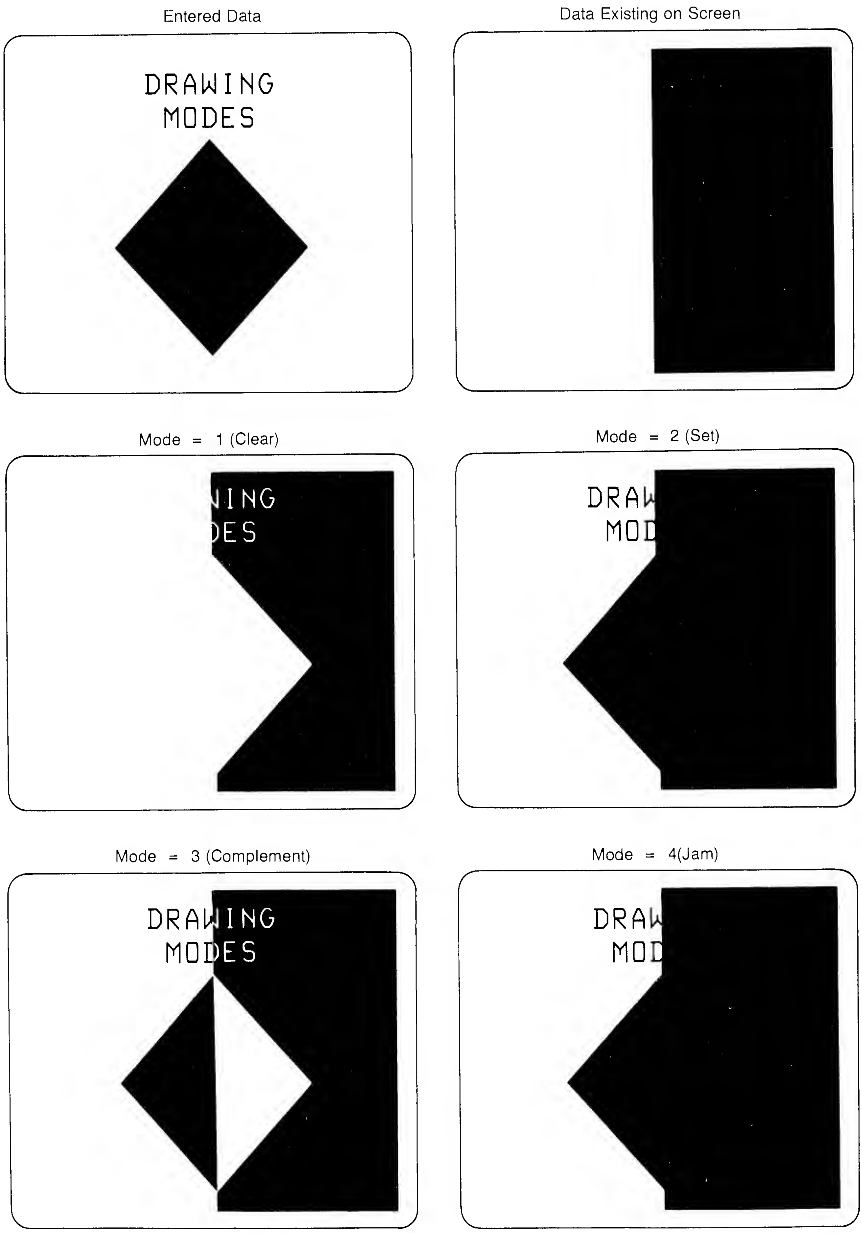


Table 2-3. Drawing Modes

Definitions: NOP - no change
exd - existing data

<x>	MODE	FUNCTION	ENTERED DATA	ACTION	EXAMPLE	
0	NOP	Picture protect, no change.	0	NOP		
			1	NOP		
1	Clear	For each 1 in the entered data, the corresponding bit in the existing data is set to 0.	0	NOP	1 1 1 1 0 0 0 0	Existing data
			1	0	1 0 1 0 1 0 1 0	Entered data
					0 1 0 1 0 0 0 0	New data
2	Set	For each 1 in the entered data, the corresponding bit in the existing data is set to 1.	0	NOP	1 1 1 1 0 0 0 0	Existing data
			1	1	1 0 1 0 1 0 1 0	Entered data
					1 1 1 1 1 0 1 0	New data
3	Comp	For each 1 in the entered data, complements the corresponding bit in the existing data.	0	NOP	1 1 1 1 0 0 0 0	Existing data
			1	NOT exd	1 0 1 0 1 0 1 0	Entered data
					0 1 0 1 1 0 1 0	New data
4	Jam	For each 1 in the entered data, the corresponding bit in the existing data is set to 1. For each 0 in the entered data, the corresponding bit in the existing data is set to 0.	0	0	1 1 1 1 0 0 0 0	Existing data
			1	1	1 0 1 0 1 0 1 0	Entered data
					1 0 1 0 1 0 1 0	New data

Point Notation for Drawing Lines

Lines are drawn by listing the coordinates of line end points in escape sequences. The following paragraphs deal with the codes used to represent point coordinates in escape sequences.

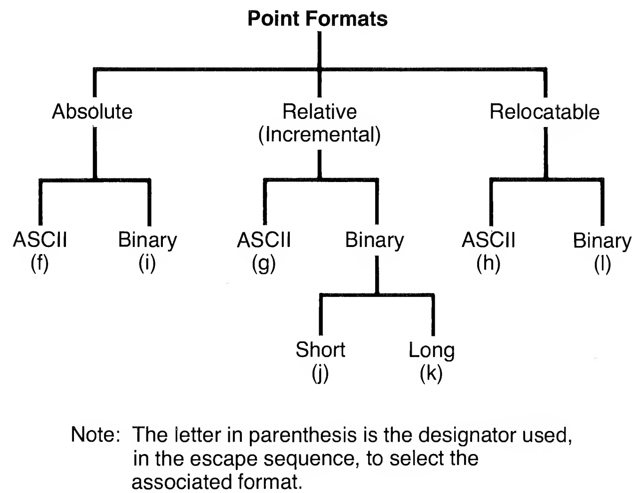
Seven formats are used to specify endpoints. These formats can be arranged into three groups, according to the reference from which the distances to the points are measured: absolute, relative (incremental), and relocatable. In each format, a point is specified as the distances, in the X and Y dimensions, of the point from the reference.

FORMAT TYPE	REFERENCE
Absolute	Origin of graphics memory (0,0)
Relative (Incremental)	Current pen position
Relocatable	Current relocatable origin

Each of the three groups can be divided into two subgroups, ASCII and binary, according to the code used to transfer the point data between the host computer and terminal. Essentially, the difference to the user is that ASCII-coded data is easy to generate and debug; binary-coded data results in more efficient data transfers. Figure 2-6 illustrates the format relationships.

If no format is specified in the graphics command, ASCII absolute format is assumed. To minimize communications overhead, more than one point can be given in a command.

Figure 2-6. Point Formats



ASCII Formats

In the ASCII formats, coordinates are specified with ASCII characters 0 through 9. This means that numeric characters generated by a simple print statement can be used to specify X,Y pairs. The first value is used as the X coordinate, and the second as the Y coordinate.

Spaces or commas must be used to delimit the X and Y values. Excess delimiters are ignored. Digits following a decimal point are ignored (i.e. 123.456 is read as 123).

Exponential notation cannot be used; coordinate values must be in integer form. The number of bytes necessary to specify a single end point depends on the magnitude of the values.

ASCII Absolute Format. The values used in ASCII absolute format can range between −16384 and 16383. Only points where X is in the range 0 to 511 (low resolution) or 0 to 639 (high resolution) and Y is in the range 0 to 389 (low

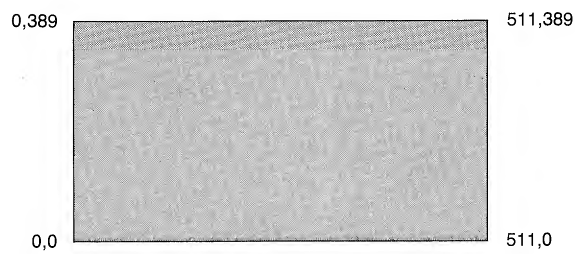
resolution) or 0 to 399 (high resolution) will be visible on the screen. The escape sequence is as follows:

```
Ec*p f
```

Example: The following example draws lines around the perimeter of the screen (low resolution screen). Figure 2-7 illustrates the result.

```
Ec*p a 0,0 511,0 511,389 0, 389 0, 0Z
```

Figure 2-7. ASCII Absolute Format Example



Since no format is indicated, ASCII absolute is assumed. The "a" raises the pen, which is moved to (0,0) and lowered. Lines are then drawn to (511,0), (511,389), (0,389), and back to (0,0).(Note that the values are delimited by spaces or commas. The upper case Z terminates the sequence. Embedded carriage return and line feed characters are ignored.

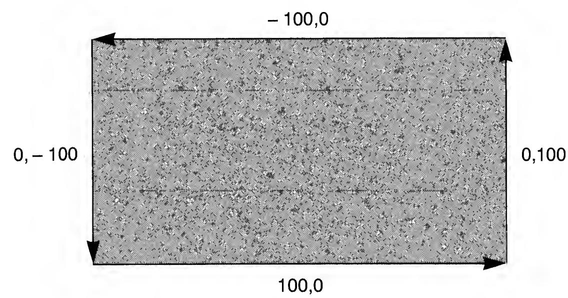
ASCII Incremental Format. In the ASCII incremental format, you can specify a delta X and a delta Y. These values are added to the current pen position to obtain a new end point. The first value is read as delta X and the second as delta Y. The escape sequence is as follows:

```
Ec*p g
```

Example: To draw a square 100 units on a side, the following sequence could be used. The resulting square is illustrated in figure 2-8.

```
Ec*p g 100,0 0,100 -100,0 0,-100 Z
```

Figure 2-8. ASCII Incremental Format Example



Beginning at the current pen position, a series of lines is drawn by moving the pen 100 units to the right, up 100 units, left 100 units, and finally down 100 units. The same figure could have been drawn at any screen location by first positioning the pen to the desired starting point before sending the drawing sequence.

ASCII Relocatable Format. After you have sent the terminal the coordinates of a relocatable origin, as specified in "Using a Relocatable Origin" in Section 3, ASCII relocatable format allows you to send point coordinates as though they were absolute coordinates.

The terminal adds the coordinates of the relocatable origin to the relocatable-formatted coordinates to determine the point on the screen where the point will be plotted. This allows plotting symbol elements, which are specified in absolute coordinates, in different screen locations, as shown in the following example. The escape sequence is as follows:

```
Ec*p h
```

Example: Draw a resistor symbol stored in absolute coordinates at screen locations 50,100 and 200,100 (figure 2-9).

`0,10 10,10 15,15`
`25,5 35,15 45,5`
`50,10 60,10`
	(Resistor data)

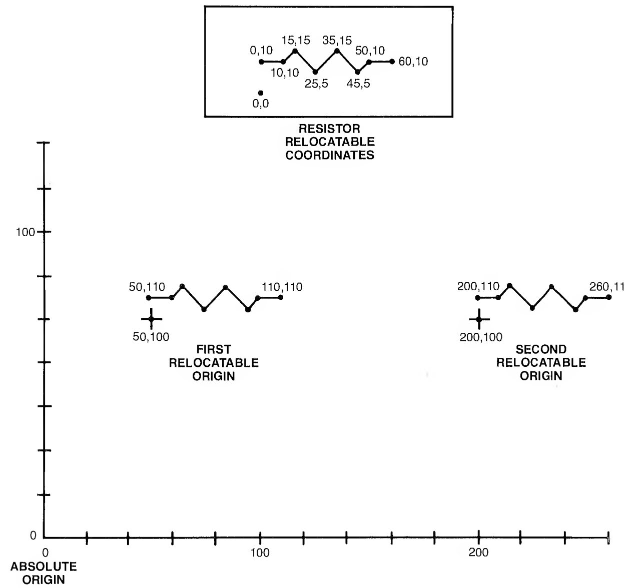
`Ec*m 50,100 J`	(Set first origin)

`Ec*p a h 0,10 10,10`
`15,15 25,5`
`35,15 45,5`
`50,10 60,10`
	(Draw first resistor)

`Ec*m 200,100 J`	(Set second origin)

`Ec*p a h 0,10 10,10`
`15,15 25,5`
`35,15 45,5`
`50,10 60,10`
	(Draw second resistor)

Figure 2-9. Relocatable Origin Example



Binary Formats

In binary formats, the coordinate values of points are specified, in the escape sequence, as ASCII characters. They are transmitted to the terminal in the form of the bit patterns of the ASCII characters, as listed in table 2-4. The number of characters required to specify an X or Y coordinate depends on the format used. The values can be from −16384 to 16383.

Note

For all binary formats, ASCII characters with decimal codes in the range 32-63 are used, in escape sequences, as code characters representing point coordinate values. Therefore, although the examples of escape sequences in this discussion use spaces (ASCII decimal code 32) for clarity, they should not be used in actual practice. Note that many other symbols (such as , / < &) also fall within the prohibited range.

Table 2-4. Binary Number/ASCII Character Conversion Chart

ASCII Character	Bit Pattern	ASCII Character	Bit Pattern
SP	01 0 0000	0	01 1 0000
!	01 0 0001	1	01 1 0001
"	01 0 0010	2	01 1 0010
#	01 0 0011	3	01 1 0011
$	01 0 0100	4	01 1 0100
%	01 0 0101	5	01 1 0101
&	01 0 0110	6	01 1 0110
"	01 0 0111	7	01 1 0111
(	01 0 1000	8	01 1 1000
)	01 0 1001	9	01 1 1001
*	01 0 1010	:	01 1 1010
+	01 0 1011	;	01 1 1011
,	01 0 1100	<	01 1 1100
-	01 0 1101	=	01 1 1101
.	01 0 1110	>	01 1 1110
/	01 0 1111	?	01 1 1111

The ASCII characters used to represent a decimal number can be selected from table 2-5.

Table 2-5. Decimal Number/ASCII Character Conversion for Binary Formats

	0	1	2	3	4	5	6	7	8	9
0	■■	■!	■"	■#	■$	■%	■&	■'	■(	■)
10	■*	■+	■,	■-	■.	■/	■0	■1	■2	■3
20	■4	■5	■6	■7	■8	■9	■:	■;	■<	■=
30	■>	■?	!■	!!	!"	!#	!$	!%	!&	!'
40	!(	!)	!*	!+	!,	!-	!.	!/	!0	!1
50	!2	!3	!4	!5	!6	!7	!8	!9	!:	!;
60	!<	!=	!>	!?	"■	"!	""	"#	"$	"%
70	"&	"'	"(	")	"*	"+	",	"-	".	"/
80	"0	"1	"2	"3	"4	"5	"6	"7	"8	"9
90	":	";	"<	"=	">	"?	#■	#!	#"	##
100	#$	#%	#&	#'	#(	#)	#*	#+	#,	#-
110	#.	#/	#0	#1	#2	#3	#4	#5	#6	#7
120	#8	#9	#:	#;	#<	#=	#>	#?	$■	$!
130	$"	$#	$$	$%	$&	$'	$(	$)	$*	$+
140	$,	$-	$.	$/	$0	$1	$2	$3	$4	$5
150	$6	$7	$8	$9	$:	$;	$<	$=	$>	$?
160	%■	%!	%"	%#	%$	%%	%&	%'	%(	%)
170	%*	%+	%,	%-	%.	%/	%0	%1	%2	%3
180	%4	%5	%6	%7	%8	%9	%:	%;	%<	%=
190	%>	%?	&■	&!	&"	&#	&$	&%	&&	&'
200	&(	&)	&*	&+	&,	&-	&.	&/	&0	&1
210	&2	&3	&4	&5	&6	&7	&8	&9	&:	&;
220	&<	&=	&>	&?	'■	'!	'"	'#	'$	'%
230	'&	''	'(	')	'*	'+	',	'-	'.	'/
240	'0	'1	'2	'3	'4	'5	'6	'7	'8	'9
250	':	';	'<	'=	'>	'?	(■	(!	("	(#
260	($	(%	(&	('	((	()	(*	(+	(,	(-
270	(.	(/	(0	(1	(2	(3	(4	(5	(6	(7
280	(8	(9	(:	(;	(<	(=	(>	(?	)■	)!
290	)"	)#	)$	)%	)&	)'	)(	))	)*	)+
300	),	)-	).	)/	)0	)1	)2	)3	)4	)5
310	)6	)7	)8	)9	):	);	)<	)=	)>	)?
320	*■	*!	*"	*#	*$	*%	*&	*'	*(	*)
330	**	*+	*,	*-	*.	*/	*0	*1	*2	*3
340	*4	*5	*6	*7	*8	*9	*:	*;	*<	*=

Note: ■ indicates a "space" character; every coordinate address must consist of the two characters shown in the table.

Table 2-5. Decimal Number/ASCII Character Conversion for Binary Formats (continued)

	0	1	2	3	4	5	6	7	8	9
350	*>	*?	+■	+!	+"	+#	+$	+%	+&	+'
360	+(	+)	+*	++	+,	+-	+.	+/	+0	+1
370	+2	+3	+4	+5	+6	+7	+8	+9	+:	+;
380	+<	+=	+>	+?	,■	,!	,"	,#	,$	,%
390	,&	,'	,(	,)	,*	,+	,,	,-	,.	,/
400	,0	,1	,2	,3	,4	,5	,6	,7	,8	,9
410	,:	,;	,<	,=	,>	,?	-■	-!	-"	-#
420	-$	-%	-&	-'	-(	-)	-*	-+	-,	--
430	-.	-/	-0	-1	-2	-3	-4	-5	-6	-7
440	-8	-9	-:	-;	-<	-=	->	-?	.■	.!
450	."	.#	.$	.%	.&	.'	.(	.)	.*	.+
460	.,	.-	..	./	.0	.1	.2	.3	.4	.5
470	.6	.7	.8	.9	.:	.;	.<	.=	.>	.?
480	/■	/!	/"	/#	/$	/%	/&	/'	/(	/)
490	/*	/+	/,	/-	/.	//	/0	/1	/2	/3
500	/4	/5	/6	/7	/8	/9	/:	/;	/<	/=
510	/>	/?	0■	0!	0"	0#	0$	0%	0&	0'
520	0(	0)	0*	0+	0,	0-	0.	0/	00	01
530	02	03	04	05	06	07	08	09	0:	0;
540	<■	0=	0>	0?	1■	1!	1"	1#	1$	1%
550	1&	1'	1(	1)	1*	1+	1,	1-	1.	1/
560	10	11	12	13	14	15	16	17	18	19
570	1:	1;	1<	1=	1>	1?	2■	2!	2"	2#
580	2$	2%	2&	2'	2(	2)	2*	2+	2,	2-
590	2.	2/	20	21	22	23	24	25	26	27
600	28	29	2:	2;	2<	2=	2>	2?	3■	3!
610	3"	3#	3$	3%	3&	3'	3(	3)	3*	3+
620	3,	3-	3.	3/	30	31	32	33	34	35
630	36	37	38	39	3:	3;	3<	3=	3>	3?
640	4■	4!	4"	4#	4$	4%	4&	4'	4(	4)
650	4*	4+	4,	4-	4.	4/	40	41	42	43
660	44	45	46	47	48	49	4:	4;	4<	4=
670	4>	4?	5■	5!	5"	5#	5$	5%	5&	5'
680	5(	5)	5*	5+	5,	5-	5.	5/	50	51
690	52	53	54	55	56	57	58	59	5:	5;
700	5<	5=	5>	5?	6■	6!	6"	6#	6$	6%
710	6&	6'	6(	6)	6*	6+	6,	6-	6.	6/

Binary Absolute Format. Binary absolute data is plotted with respect to an origin at 0,0. Four bytes are required to specify a single end point. A 10 bit coordinate in the range 0–1023, is sent for both x and y.

The bytes are ordered as follows:

BIT	7	6	5	4	3	2	1	
BYTE 1	0	1	X9	X8	X7	X6	X5	HI X
BYTE 2	0	1	X4	X3	X2	X1	X0	LOW X
BYTE 3	0	1	Y9	Y8	Y7	Y6	Y5	HI Y
BYTE 4	0	1	Y4	Y3	Y2	Y1	Y0	LOW Y

Although it is possible to send coordinates in the range 0 to 1023 , only points in the range 0–511 for X and 0–389 for Y (or, for high resolution, 0–639 for X and 0–399 for Y) are visible on the screen. Lines extending off the screen are clipped. If the data requires scaling, this must be done before the data is sent to the terminal. The escape sequence used is:

```
Ec*p i
```

Example:

The following example shows how the 4 data bytes are computed. The numbers are converted to the 10 bit binary equivalent. Bits 7 and 6 are set to 01 to indicate a parameter.

```
X = 0 =00000  00000          Y = 0 =00000  00000
       HI X   LOW X                  HI Y   LOW Y
```

```
BYTE 1 =  01  00000 = SPACE   HI X
BYTE 2 =  01  00000 = SPACE   LOW X

BYTE 3 =  01  00000 = SPACE   HI Y
BYTE 4 =  01  00000 = SPACE   LOW Y
```

```
X=360= 01011  01000        Y=180= 00101  10100
       HI X   LOW X               HI Y   LOW Y

       BYTE 1 = 01 01011 = +  HI X
       BYTE 2 = 01 01000 = (  LOW X

       BYTE 3 = 01 00101 = %  HI Y
       BYTE 4 = 01 10100 = 4  LOW Y
```

Example:

An escape sequence to draw a line from 0,0 to 360,180 is as follows. (Although spaces are used in the following escape sequence for clarity, they are illegal in actual use, because spaces have meaning when specifying coordinates.)

```
Ec*p i a   SP SP   SP SP    +(      %4      Z
             X=0     Y=0   X=360   Y=180
```

"Ec*p" selects a plotting sequence. The "i" specifies binary absolute format. The "a" raises the pen up. The first four bytes (all spaces) move the raised pen to 0,0 where it is lowered. The next four bytes specify the point 360,180. After the fourth byte is received, the pen is moved to that point, drawing a line. The upper case "Z" terminates the escape sequence.

Binary Short Incremental Format. The short incremental format uses two bytes to specify a delta X and a delta Y in the range −16 to +15. The five least significant bits are interpreted as a signed, two's complement number. This number is added to the current pen position to obtain the new end point. The escape sequence used is:

```
Ec*p j
```

The data bytes are ordered as follows:

BIT	7	6	5	4	3	2	1
BYTE 1	0	1	<		DELTA X		>
BYTE 2	0	1	<		DELTA Y		>

For convenience, table 2-6 lists the ASCII characters to be used to represent coordinates in binary short incremental format.

Table 2-6. Coordinate Number/ASCII Code Conversion for Use in Binary Short Incremental Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

Example:

The following example illustrates the computation and use of short incremental format:

```
DELTA X = -12 = 10100    DELTA Y = 6 = 00110

  BYTE1 = 01 10100 = 4   DELTA X
  BYTE2 = 01 00110 = &   DELTA Y
```

The following sequence moves the pen to `360, 180` in absolute format, then draws a line to X = `360-12` = `348`, Y = `180+6` = `186`.

Although spaces are used for clarity in the following sequence, spaces should not be used in actual practice, except as code for the digit "0".

```
Ec*p i a + ( % 4 j 4 & Z
```

Binary Incremental Format. Incremental is similar to short incremental, but with a larger range. Using six bytes, delta X and Y can range from −16384 to +16383.

BIT	7	6	5	4	3	2	1	
BYTE1	0	1	DX14	DX13	DX12	DX11	DX10	HI DELTA X
BYTE2	0	1	DX9	DX8	DX7	DX6	DX5	MID DELTA X
BYTE3	0	1	DX4	DX3	DX2	DX1	DX0	LOW DELTA X
BYTE4	0	1	DY14	DY13	DY12	DY11	DY10	HI DELTA Y
BYTE5	0	1	DY9	DY8	DY7	DY6	DY5	MID DELTA Y
BYTE6	0	1	DY4	DY3	DY2	DY1	DY0	LOW DELTA Y

The escape sequence used for binary incremental format is:

```
Ec*p k
```

Example:

The following example shows how incremental data bytes are generated.

```
DELTA X = -400 = 11111   10011    10000
                 HI DX   MID DX   LO DX

DELTA Y = 100 =  00000   00011    00100
                 HI DY   MID DY   LO DY

BYTE 1 = 01 11111 = ?          HI DELTA X
BYTE 2 = 01 10011 = 3         MID DELTA X
BYTE 3 = 01 10000 = 0          LO DELTA X

BYTE 4 = 01 00000 = space      HI DELTA Y
BYTE 5 = 01 00011 = #         MID DELTA Y
BYTE 6 = 01 00100 = $          LO DELTA Y
```

Binary Relocatable Format. Binary relocatable format, like ASCII relocatable format, allows use of a relocatable origin, to which relocatable-formatted point coordinates are referenced.

A relocatable origin is implemented by first sending the coordinates of the selected origin to the terminal as described in "Using a Relocatable Origin". With the new origin established, the terminal locates any point coordinates it receives, which are formatted in a relocatable format, with respect to the relocatable origin. The terminal computes the actual screen address by adding the relocatable origin to the X,Y pair. The escape code used to specify binary relocatable format is:

```
Ec*p 1
```

Binary relocatable format specifies absolute X and Y coordinates in the range −16384 to +16383 using six bytes. The value for the 0,0 point is sent to the terminal in a relocatable origin escape sequence.

BIT	7	6	5	4	3	2	1	
BYTE 1	0	1	X14	X13	X12	X11	X10	HI X
BYTE 2	0	1	X9	X8	X7	X6	X5	MID X
BYTE 3	0	1	X4	X3	X2	X1	X0	LOW X
BYTE 4	0	1	Y14	Y13	Y12	Y11	Y10	HI Y
BYTE 5	0	1	Y9	Y8	Y7	Y6	Y5	MID Y
BYTE 6	0	1	Y4	Y3	Y2	Y1	Y0	LOW Y

Example:

The following example shows how relocatable data bytes are computed.

RELOC X = -600 =	11111	01101	01000
	HI X	MID X	LOW X
RELOC Y = 200 =	00000	00110	01000
	HI Y	MID Y	LOW Y

BYTE 1 = `01 11111` = `?`		HI X
BYTE 2 = `01 01101` = `-`		MID X
BYTE 3 = `01 01000` = `(`		LOW X
BYTE 4 = `01 00000` = `space`		HI Y
BYTE 5 = `01 00110` = `&`		MID Y
BYTE 6 = `01 01000` = `(`		LOW Y

Mixing Formats

There are no restrictions on mixing formats; simply specify the new format to be used and follow it with new data in the new format. By restricting data values to binary values between 32 and 63 (the printing graphics characters and numbers), the plotting commands "a" through "z" can be intermixed in the binary data.

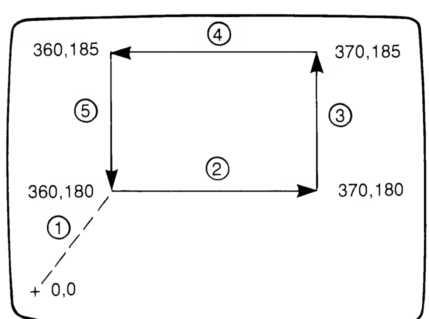
Example:

Move the pen to `360,180` in ASCII absolute format, then draw a box 10 units wide by five units high, using binary short incremental format (figure 2-10).

Although spaces are used, for clarity, in the following escape sequence, they should not appear in the sequence actually used, except as code for a "0" digit.

```
Ec*p a  f 360,180 j * SP SP % 6 SP SP ; Z
```

Figure 2-10. Example of Mixed Data Formats



COLOR

Color Notation

The terminal uses two color notation systems: HSL (hue, saturation, and luminosity) and RGB (red, green, and blue). Each system allows specifying a color and the vividness of the color.

HSL Method. The HSL method breaks color down into three components:

- Hue. The mix of light frequencies (what we usually call color). It's range is from 0 to 1, as shown below:

0.00 - 0.04	Red	0.46 - 0.54	Cyan
0.05 - 0.08		0.55 - 0.58	
0.09 - 0.12		0.59 - 0.62	
0.13 - 0.20	Yellow	0.63 - 0.70	Blue
0.21 - 0.25		0.71 - 0.74	
0.26 - 0.29		0.75 - 0.79	
0.30 - 0.37	Green	0.80 - 0.87	Magenta
0.38 - 0.41		0.88 - 0.91	
0.42 - 0.45		0.92 - 0.95	
		0.96 - 1.00	Red

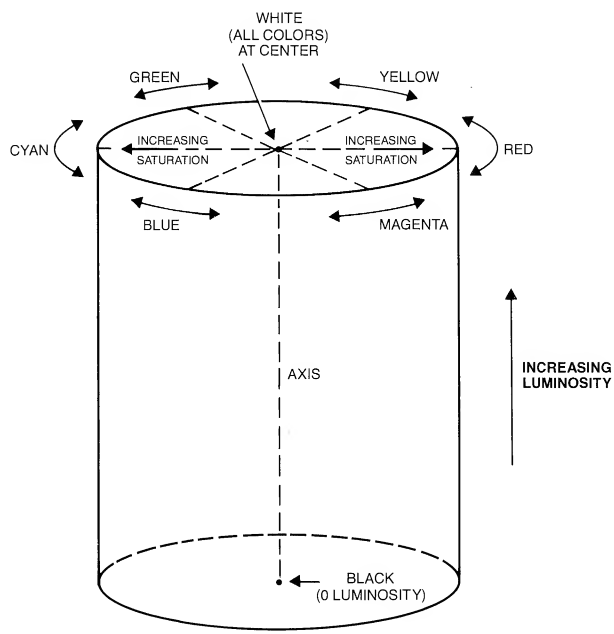
- Saturation. Saturation can range from 0 (white) to 1 (maximum intensity of the color selected by the hue).

- Luminosity. Selects the amount of light applied to the hue. It ranges from 0 (black - no light) to 1 (maximum light).

The concepts of hue, saturation and luminosity are illustrated in figure 2-11 as a HSL color cylinder. Hue goes circularly around the cylinder. Saturation increases from 0 (white), at the axis of the cylinder, to 1, which is maximum intensity of the color, at the outside edge of the cylinder. Luminosity ranges from 0 (black), at the bottom of the cyl-

inder, to 1 (maximum luminosity or light) at the top of the cylinder.

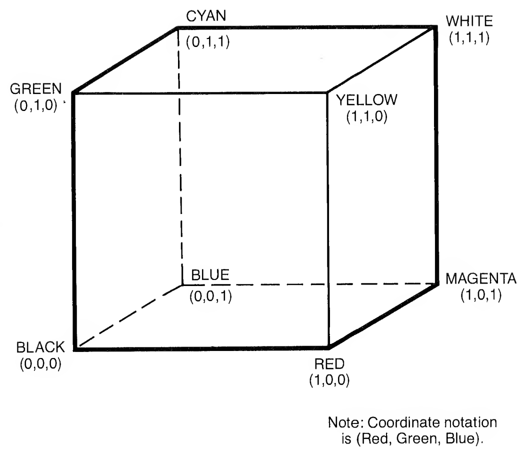
Figure 2-11. HSL Color Cylinder



RGB Method. The RGB method mixes three colors, red, green, and blue, to derive other colors. For example, yellow is produced by mixing red and green; cyan by mixing green and blue; and magenta by mixing red and blue. The numerical values for red, green, and blue range from 0.00 to 1.00.

Figure 2-12 illustrates the RGB concepts in the form of a color cube. Each corner of the cube is a color, red, green, blue, cyan, magenta, yellow, white (all colors), and black (no color). Moving along one edge of the cube from one corner to another, the color for the first corner gradually blends into the color for the next corner.

Figure 2-12. RGB Color Cube



You can select intensity values for any of the colors red, green and blue. The selection range is 0.00 to 1.00. However, any intensity selection is converted, by the terminal to one of four selections: 0.00, 0.33, 0.66, and 1.00. Table 2-7 lists the entered and used intensity values.

Table 2-7. Selectable and Used RGB Color Intensity Values.

SELECTED VALUE	0.00–0.24	0.25–0.49	0.50–0.74	0.75–1.00
USED VALUE	0.00	0.33	0.66	1.00

The mix of intensity values for the three primary colors determines the resultant color to be displayed on the screen. With four possible intensity values for each of the three primary colors, the result is a total of 64 possible colors for display.

Color Notation Conversion

Color values expressed in RGB notation can be converted to HSL notation, and vice versa. Table 2-8 converts from RGB to HSL; table 2-9 converts from HSL to RGB.

Table 2-8. RGB to HSL Conversion

RGB CONDITION	CONVERSION
R=G=B	H=0 S=0 L=R(=G=B)
B<=G<=R	H=(1-(R-G)/(R-B))/6 S=(R-B)/R L=R
B<=R<=G	H=(1+(G-R)/(G-B))/6 S=(G-B)/G L=G
G<=B<=R	H=(5+(R-B)/(R-G))/6 S=(R-G)/R L=R
G<=R<=B	H=(5-(B-R)/(B-G))/6 S=(B-G)/B L=B
R<=B<=G	H=(3-(G-B)/(G-R))/6 S=(G-R)/G L=G
R<=G<=B	H=(3+(B-G)/(B-R))/6 S=(B-R)/B L=B

Table 2-9. HSL to RGB Conversion

Definitions:

INTEGER(xx) means: use the integer portion of xx.

Sextant=INTEGER(6*H)

Fraction=(6*H)-Sextant

X=L*(1-S)

Y=L*(1-(S*Fraction))

Z=L*(1-(S*(1-Fraction)))

SEXTANT	CONVERSION
0 or 6	R=L G=Z B=X
1	R=Y G=L B=X
2	R=X G=L B=Z
3	R=X G=Y B=L
4	R=Z G=X B=L
5	R=L G=X B=Y

Intensity Values for the Color Spectrum

Table 2-10 lists the intensity values, in the RGB system, for all available colors. The equivalent values for the HSL system are also listed.

Table 2-10. Color Intensity Values

R	G	B	H	S	L	
0.00	0.00	0.00	0.00	0.00	0.00	BLACK
0.00	0.00	0.33	0.67	1.00	0.33	
0.00	0.00	0.66	0.67	1.00	0.66	
0.00	0.00	1.00	0.67	1.00	1.00	BLUE
0.00	0.33	0.00	0.33	1.00	0.33	
0.00	0.33	0.33	0.50	1.00	0.33	
0.00	0.33	0.66	0.58	1.00	0.66	
0.00	0.33	1.00	0.61	1.00	1.00	
0.00	0.66	0.00	0.33	1.00	0.66	
0.00	0.66	0.33	0.42	1.00	0.66	
0.00	0.66	0.66	0.50	1.00	0.66	
0.00	0.66	1.00	0.56	1.00	1.00	
0.00	1.00	0.00	0.33	1.00	1.00	GREEN
0.00	1.00	0.33	0.39	1.00	1.00	
0.00	1.00	0.66	0.44	1.00	1.00	
0.00	1.00	1.00	0.50	1.00	1.00	CYAN
0.33	0.00	0.00	1.00	1.00	0.33	
0.33	0.00	0.33	0.83	1.00	0.33	
0.33	0.00	0.66	0.75	1.00	0.66	
0.33	0.00	1.00	0.72	1.00	1.00	
0.33	0.33	0.00	0.17	1.00	0.33	
0.33	0.33	0.33	0.00	0.00	0.33	
0.33	0.33	0.66	0.67	0.50	0.66	
0.33	0.33	1.00	0.67	0.67	1.00	
0.33	0.66	0.00	0.25	1.00	0.66	
0.33	0.66	0.33	0.33	0.50	0.66	
0.33	0.66	0.66	0.50	0.50	0.66	
0.33	0.66	1.00	0.58	0.67	1.00	
0.33	1.00	0.00	0.28	1.00	1.00	
0.33	1.00	0.33	0.33	0.67	1.00	
0.33	1.00	0.66	0.42	0.67	1.00	
0.33	1.00	1.00	0.50	0.67	1.00	

Table 2-10. Color Intensity Values (continued)

R	G	B	H	S	L	
0.66	0.00	0.00	1.00	1.00	0.66	
0.66	0.00	0.33	0.92	1.00	0.66	
0.66	0.00	0.66	0.83	1.00	0.66	
0.66	0.00	1.00	0.78	1.00	1.00	
0.66	0.33	0.00	0.08	1.00	0.66	
0.66	0.33	0.33	1.00	0.50	0.66	
0.66	0.33	0.66	0.83	0.50	0.66	
0.66	0.33	1.00	0.75	0.67	1.00	
0.66	0.66	0.00	0.17	1.00	0.66	
0.66	0.66	0.33	0.17	0.50	0.66	
0.66	0.66	0.66	0.00	0.00	0.66	
0.66	0.66	1.00	0.67	0.34	1.00	
0.66	1.00	0.00	0.22	1.00	1.00	
0.66	1.00	0.33	0.25	0.67	1.00	
0.66	1.00	0.66	0.33	0.34	1.00	
0.66	1.00	1.00	0.50	0.34	1.00	
1.00	0.00	0.00	1.00	1.00	1.00	RED
1.00	0.00	0.33	0.95	1.00	1.00	
1.00	0.00	0.66	0.89	1.00	1.00	
1.00	0.00	1.00	0.83	1.00	1.00	MAGENTA
1.00	0.33	0.00	0.06	1.00	1.00	
1.00	0.33	0.33	1.00	0.67	1.00	
1.00	0.33	0.66	0.92	0.67	1.00	
1.00	0.33	1.00	0.83	0.67	1.00	
1.00	0.66	0.00	0.11	1.00	1.00	
1.00	0.66	0.33	0.08	0.67	1.00	
1.00	0.66	0.66	1.00	0.34	1.00	
1.00	0.66	1.00	0.83	0.34	1.00	
1.00	1.00	0.00	0.17	1.00	1.00	YELLOW
1.00	1.00	0.33	0.17	0.67	1.00	
1.00	1.00	0.66	0.17	0.34	1.00	
1.00	1.00	1.00	0.00	0.00	1.00	WHITE

Color Tools

The color terminal uses the concepts of palettes and pens.

Pens. As described previously, a pen is used to draw points and lines and to fill areas. It has two states: up and down. It can be used to draw only while in the down state. The up state is used for moving the pen without drawing a line. A pen is assigned a color from a palette of eight colors.

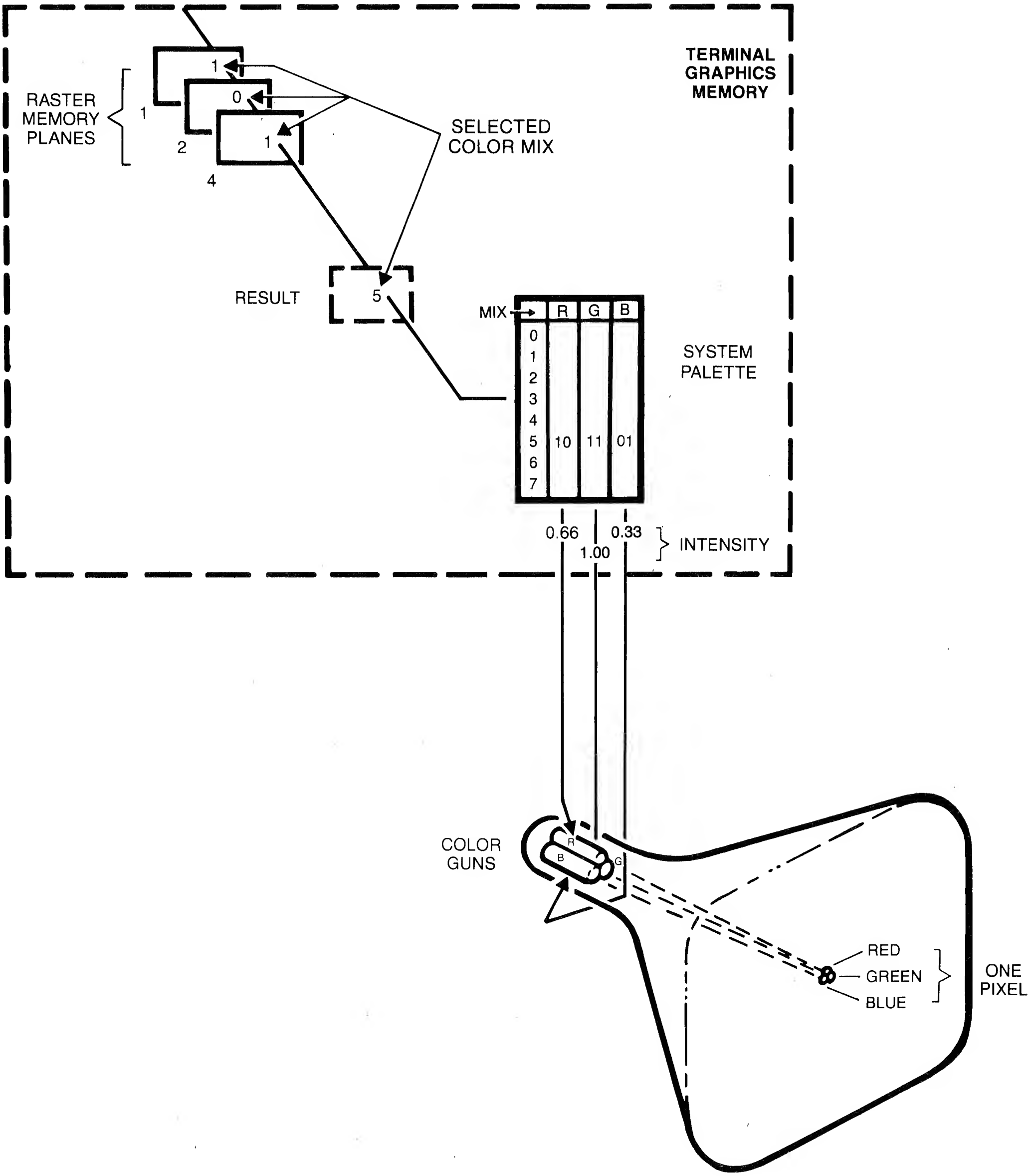
The terminal uses five types of pens to create pictures: the primary pen, the secondary pen, the background pen, the area boundary pen, and the graphics text pen. The primary pen is the one used to draw points and lines. The secondary pen is used, in some drawing modes, to fill areas or draw patterned lines. The color assigned the background pen is the color of the picture background. The area boundary pen is a pen which can be used to draw outlines around area fills. The graphics text pen is used for creating graphics text.

Palettes. A palette consists of eight color mixes, 0–7 (figure 2-13). Each mix is composed of the three primary colors, red, green, and blue. Each primary color is assigned an intensity level (0, 1, 2, or 3 — or 0.00, 0.33, 0.66, or 1.00). The intensity level for each primary color determines the intensity of the electron beam from the associated color gun (red, green, or blue) in the cathode ray tube (CRT). With four intensity levels for each of three color guns, 64 colors can be produced for each screen pixel.

When an intensity level has been assigned to each of the colors red, green, and blue for a given color mix, the mix has, effectively, been assigned one of the 64 colors. When all eight mixes on a palette have been assigned colors, the palette is said to be defined.

Up to 16 defined palettes can exist in terminal memory at a given time. Each palette is assigned a number from 0 to 15, with 0 as the system palette. The system palette is the palette from which the colors displayed on the screen are taken.

Figure 2-13. Color Selection for a Pixel



The colors from any defined palette can be loaded into the system palette at any time. When this is done, the colors on the screen change to the new colors assigned to the system palette. This occurs because the points, lines, and area fills which compose the screen picture are each associated with a mix number (0–7) on the system palette. When the color assigned a given mix on the system palette changes, all items on the screen which were drawn using that mix change to the new color assigned to the mix.

The primary, secondary, background, area boundary, and graphics text pens are assigned mixes from the system palette.

How Color is Generated

The Cathode Ray Tube (CRT). The face, or screen, of a color terminal CRT is coated with three different phosphor coatings: red, green, and blue. The CRT contains three electron guns, one for each of the phosphor coatings. A mask is used to ensure that the electron beam from the red gun hits only red phosphor dots; the beam from the green gun, only green dots; and the beam from the blue gun, only blue dots.

A group of three adjacent dots, consisting of one red, one green, and one blue dot is called a picture element (pixel). By lighting different combinations of the three dots of a pixel, eight colors can be produced: black, red, green, blue, yellow, cyan, magenta, and white. However, each dot can be lighted with one of four levels of intensity (0.00, 0.33, 0.66, and 1.00). The result is that 64 colors can be produced for each pixel.

Raster Memory. The color terminal stores graphics images to be displayed on the screen in an area of graphics memory called raster memory. Raster memory is made up of three memory planes. Each screen pixel is represented on each memory plane by a binary bit. The three memory planes are assigned binary weights of 1, 2, and 4. The three bits on the memory planes form a binary number (0–7) which selects the color mix on the system palette.

System Palette. For a given pixel, the values for all planes with a "1" stored for that pixel are summed to select the "mix" which defines the color with which the pixel will be drawn. The mix number can range from 0 to 7.

The system palette stores intensity values for the RGB color notation system. If the HSL system is being used, the HSL values are converted to RGB values before being stored on the system palette.

For each pixel, the intensity for each of the red, green, and blue colors determines the intensity of the electron beam from the associated color gun to produce the selected color for the pixel.

To produce the picture on the screen, raster memory is constantly scanned at a fixed rate, producing the color for each pixel as outlined above.

Drawing Modes for Color Terminals

The color terminal has eight drawing modes. These modes determine the way data is displayed on the screen, so that changing the drawing mode can radically change the screen appearance of a given set of data. The modes are:

NO.	NAME	NO.	NAME
0	Picture Protect	4	Jam2
1	Clear1	5	OR
2	Jam1	6	Complement2
3	Complement1	7	Clear2

The escape sequence for selecting the drawing mode is as follows:

```
Ec*m <x>a
```

where <x>, the selected mode, and the mix with which binary 0's and 1's in the entered data will be drawn are listed in table 2-12.

Table 2-11. Drawing Modes for Color Terminals

Definitions:

bmix	—	mix no. assigned background pen.
cmix	—	mix no. assigned current pen.
dmix	—	mix no. with which existing data was written.
current pen	—	primary pen, text pen, or, for dithering, the pen selected for drawing the current pixel within the 4 X 4 pixel dither pattern.
smix	—	mix no. assigned secondary pen.

MODE NO.	MODE NAME	FUNCTION	DATA BIT	RESULTING COLOR	EXAMPLE	
0	NOP	Picture protect, no change.	0 1	NOP NOP		
1	Clear1	For 1's in entered data, overwrites existing data with background pen.	0 1	NOP bmix	4	Mix no. of background pen
					3 3 3 3 0 0 0 0	Mix no.'s of existing data
					1 0 1 0 1 0 1 0	Entered data (binary)
					4 3 4 3 4 0 4 0	Mix no.'s of new data
2	Jam1	Default mode. For 1's in entered data, overwrites existing data with current pen.	0 1	NOP cmix	5	Mix no. of current pen
					3 3 3 3 0 0 0 0	Mix no.'s of existing data
					1 0 1 0 1 0 1 0	Entered data (binary)
					5 3 5 3 5 0 5 0	Mix no.'s of new data

Table 2-11. Drawing Modes for Color Terminals (continued)

MODE NO.	MODE NAME	FUNCTION	DATA BIT	RESULTING COLOR	EXAMPLE	
3	Compl	For 1's in entered data, overwrites existing data with the NOT existing data mix no. Redrawing the line a second time restores the original line.	0 1	NOP NOT dmix	3 = 011 0 = 000	Mix no.'s of existing data
					100 = 4 111 = 7	NOT 3 = 4 NOT 0 = 7
					3 3 3 3 0 0 0 0	Mix no.'s of existing data
					1 0 1 0 1 0 1 0	Entered data (binary)
					4 3 4 3 7 0 7 0	Mix no.'s of new data
4	Jam2	For 1's in entered data, overwrites existing data with current pen. For 0's in entered data, overwrites existing data with secondary pen.	0 1	smix cmix	2	Mix no. of secondary pen
					6	Mix no. of current pen
					3 3 3 3 0 0 0 0	Mix no.'s of existing data
					1 0 1 0 1 0 1 0	Entered data (binary)
					6 2 6 2 6 2 6 2	Mix no.'s of new data

Table 2-11. Drawing Modes for Color Terminals (continued)

MODE NO.	 NAME	 FUNCTION	DATA BIT	RESULTING COLOR	EXAMPLE	
5	OR	For 1's in entered data, overwrites existing data with the mix no. resulting from ORing the existing data mix no. with the mix no. of the current pen.	0 1	NOP dmix OR cmix	3 = 011	Mix no. of existing data
					1 = 001	Mix no. of current pen
					011 = 3	OR 3 and 1. New data mix no. is 3
					0 = 000	Mix no. of existing data
					1 = 001	Mix no. of current pen
					001 = 1	OR 0 and 1. New data mix no. is 1
					3 3 3 3 0 0 0 0	Mix no.'s of existing data
					1 0 1 0 1 0 1 0	Entered data (binary)
					3 3 3 3 1 0 1 0	Mix no. of new data

Table 2-11. Drawing Modes for Color Terminals (continued)

MODE NO.	MODE NAME	FUNCTION	DATA BIT	RESULTING COLOR	EXAMPLE	
6	Comp2	For 1's in entered data, overwrites existing data with the mix no. resulting from XORing the current pen mix no. with the background pen mix no., then XORing the result with the mix no. of the existing data. Redrawing the line a second time restores the original line.	0	NOP	1 = 001	Mix no. of current pen
			1	dmix		
			1	cmix XOR bmix XOR dmix	6 = 110	Mix no. of background pen
					111 = 7	XOR two mixes
					3 = 011	Mix no. of existing data
					100 = 4	XOR two pens. New data pen mix no. is 4
					111 = 7	XOR current and background pen mix no
					0 = 000	Mix no.'s of existing data
					111	XOR two mixes. New data mix no. is 7
					3 3 3 3 0 0 0 0	Mix no.'s of existing data
					1 0 1 0 1 0 1 0	Entered data (binary)
					4 3 4 3 7 0 7 0	Mix no.'s of new data

Table 2-11. Drawing Modes for Color Terminals (continued)

MODE NO.	MODE NAME	FUNCTION	DATA BIT	RESULTING COLOR	EXAMPLE	
7	Clear2	For 1's in entered data, overwrites existing data with the result of ANDing the existing data mix no. and NOT the mix no. of the current pen.	0	NOP	1 = 001	Current pen mix no.
			1	(dmix) AND (NOT cmix)	110 = 6	NOT current pen mix no.
					3 = 011	Mix no. of existing data
					010 = 2	AND two values. New data mix no. is 2
					110 = 6	NOT current pen mix no
					0 = 000	Mix no. of existing data
					000 = 0	AND two values. New data mix no. is 0
					3 3 3 3 0 0 0 0	Mix no.'s of existing data
					1 0 1 0 1 0 1 0	Entered data (binary)
					2 3 2 3 0 0 0 0	Mix no.'s of new data

Several definitions are required for the following discussion:

- Entered data — Data entered into the terminal.
- Existing data — Data existing on the screen when the entered data is entered.
- New data — Data which results from the entered and the existing data.

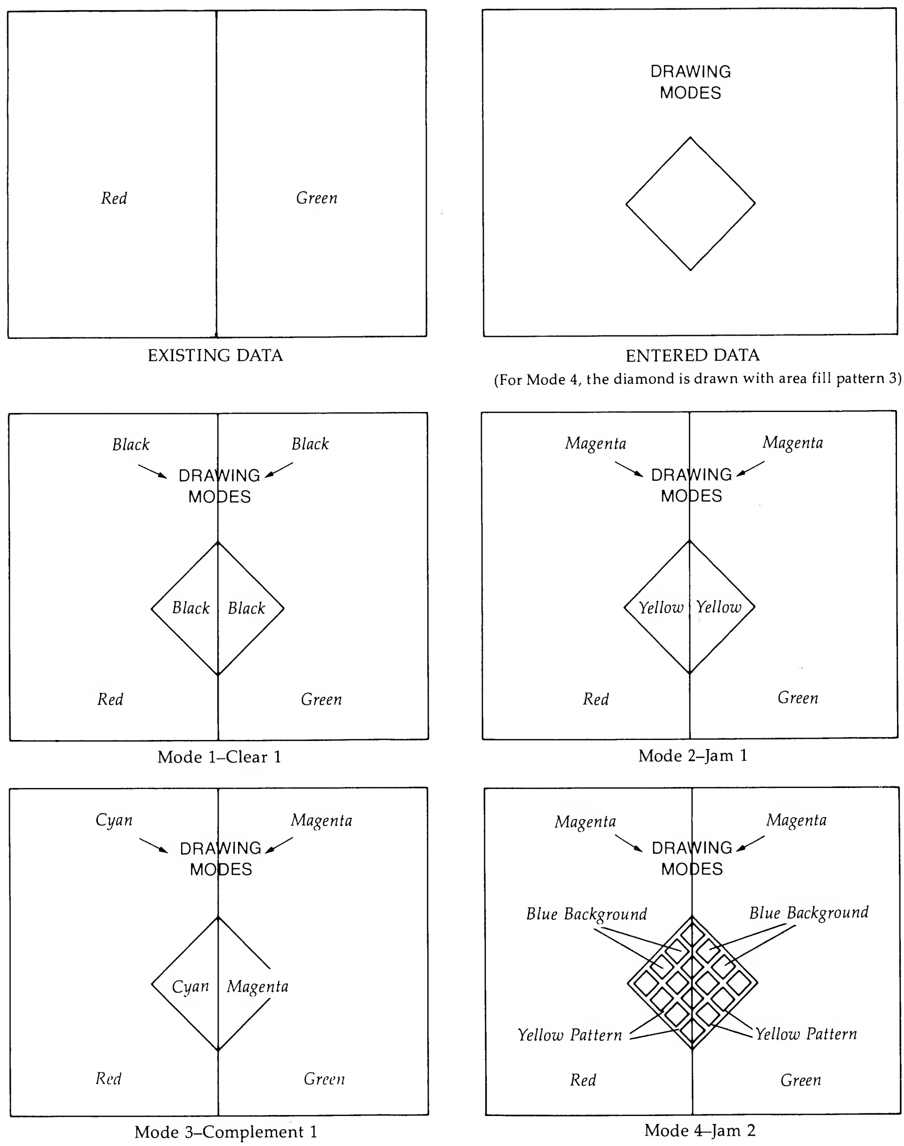
In modes 1 through 7, a 1 bit in the entered data is a signal to the terminal to act on the corresponding bit in the existing data. For 0 bits in the entered data, the existing data is not affected, except in mode 4.

New data is drawn with a color mix number selected using the following items: the mix number with which the data existing on the screen was drawn, the mix number of the current pen, background pen, or secondary pen, or a combination of these numbers.

Figure 2-14 illustrates the effect of each mode on a given set of data.

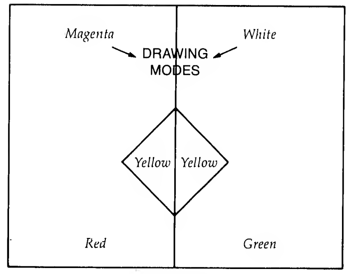
The default mode, which is selected after a power on or hard reset, is mode 2 (Jam1 mode). In this mode, "1's" are drawn with the current pen; "0's" have no effect.

Figure 2-14. Effect of Each Drawing Mode on a Given Set of Data (Color)

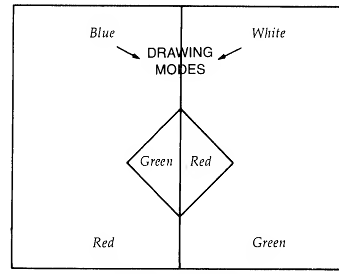


Note: Pen/ mix selections: Background pen = mix 0 (black). Secondary pen = mix 4 (blue).
Primary pen = mix 3 (yellow). Text pen = mix 5 (magenta).

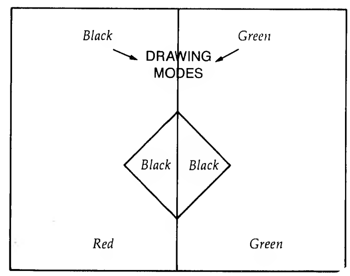
Figure 2-14. Effect of Each Drawing Mode on a Given Set of Data (continued)



Mode 5–OR



Mode 6–Complement 2



Mode 7–Clear 2

3 Graphics Operations

Introduction

This section contains a description of the terminal's graphics functions and how to use them. The information and the examples are intended for use in developing programs to create and print out screen displays. Additional information on how to use the graphics features from the keyboard is contained in the User's manual.

The contents of this section are:

- Drawing lines.
- Filling areas.
- Using Graphics text.
- Obtaining a hardcopy.
- Using a relocatable origin.
- Graphics status.
- Selecting the graphics default parameters.
- Inserting delays in graphics operations.
- Display Functions mode.
- Compatibility mode.

Drawing Lines

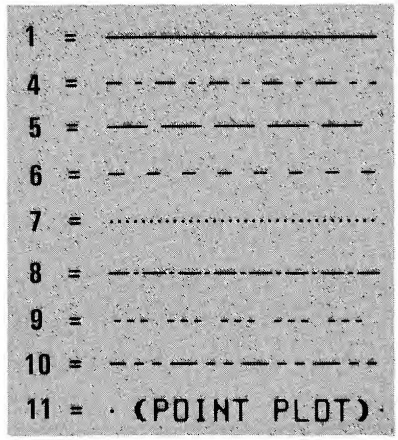
To draw any line, the line type is selected, then drawn, using separate escape sequences.

Selecting the Line Type

Once a line type is selected, it is used until another type is selected. The default line type (solid line) is selected at power on or after a hard reset. The line types are as follows:

- Isolated Points. Points are included as a special line type. Point plotting can be useful for generating "scattergram" type graphs.
- Solid Lines. A solid line is the default selection, which occurs after power on or a hard reset.
- Predefined Line Types. One of seven predefined line types (figure 3-1) can be selected.
- User-Defined Line Types. As an alternative to solid and predefined line types, you can define your own line pattern. (Refer to "Defining Your Own Line Type", later in this section.)
- Area Pattern as a Line Type. The currently selected area fill pattern can be used as a line type for drawing horizontal and vertical lines. (Refer to "Using an Area Fill Pattern as a Line Type", later in this section.)

Figure 3-1. Predefined Line Types



To select the line type, use the following escape sequence:

Ec*m <type> b

where <type> is one of the following numbers:

NO.	LINE TYPE
1	Solid line (default).
2	User defined line pattern.
3	Current area fill pattern.
4	Predefined pattern 1.
5	Predefined pattern 2.
6	Predefined pattern 3.
7	Predefined pattern 4.
8	Predefined pattern 5.

(continued)

NO.	LINE TYPE
9	Predefined pattern 6.
10	Predefined pattern 7.
11	Point plot.

Drawing the Line

To draw any line, the pen is moved, in the down state, from the start point of the line to the end point. If the current pen position isn't the start point of the line, the pen must be moved, in the up state, to the start point. Refer to "Pen Control" for pen control instructions.

The pen can be moved in either the up or down state with the following escape sequence:

```
Ec*p <format> <pen state> <x,y>
     <x,y>......<x,y> Z
```

where:

`<format>` The format in which the coordinates for
the `<x,y>` points are specified (these formats are described in Section 2):

f = ASCII absolute (default)
g = ASCII Incremental
h = ASCII relocatable
i = binary absolute
j = binary short incremental
k = binary long incremental
l = binary relocatable

`<pen state>` Up or down state of the pen:

a = up
b = down

`<x,y>`	The x and y coordinates of a point. The first coordinate is assumed to be the x coordinate. As many points as desired can be entered. Carriage return and line feed characters are ignored in the escape sequence. For ASCII formats, the coordinates can be separated by any non-letter, non-digit ASCII character, such as a comma, space, colon, etc. This is not true for binary formats (refer to "Binary Formats" in Section 2.
`Z`	Terminates the escape sequence.

Neither the `<format>` nor the `<pen state>` need be included in the escape sequence. If `<format>` isn't included, the default choice, ASCII absolute, is used. Refer to "Point Notation for Drawing Lines" in Section 2 for format information.

If `<pen state>` isn't included, the current state of the pen is used to move between the current pen position and the first point indicated in the escape sequence. If more than one point is supplied in the sequence, without an intervening pen up command, subsequent pen movements are made with the pen down (lines are drawn), because the pen is always left in the down state after a pen movement.

Note that if a parameter byte is lost or garbled in transmission, all following end points will be improperly read. To minimize data errors caused by the loss of a data byte, short sequences can be used to reset the parameter count and restore synchronization. NOPs (z), redundant format, or pen down commands can also be inserted to insure synchronization if necessary.

Graphics sequences can extend indefinitely. In general, longer sequences are preferred as they minimize the overhead necessary for a plot sequence. `Ec*p <format>`

must be sent for each series of lines. As the sequence length decreases, the percentage of preamble characters increases, and the line drawing rate goes down. The worst possible case would be to send `Ec*p <format>` for each line; approximately 50% of the characters sent would be overhead, reducing line speed by a factor of 2.

Pen Control

The terminal uses the concept of a "pen" in drawing points and lines and filling areas. The pen can be lifted (up state) or lowered (down state). It can also be positioned using absolute or relative coordinates.

After power on or a hard reset, the pen is in the down state and positioned at coordinates 0,0. After any pen movement, the pen is left in the lowered state.

Moving the pen while it is in the down state draws a line. Typical actions for drawing a line are:

1. Lift the pen.
2. Move the pen to the start point of the line.
3. Move the pen to the end point of the line.

The pen is lifted and lowered using the following escape sequences:

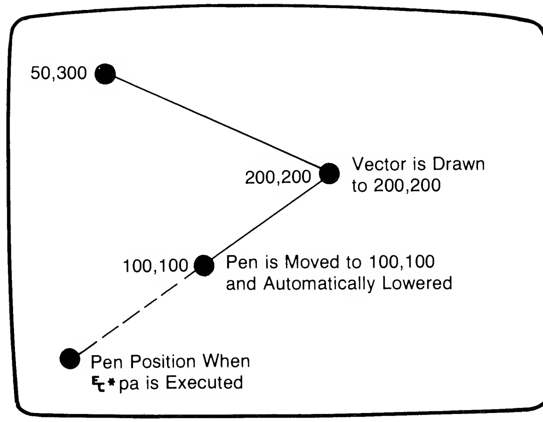
Lift Pen: `Ec*p a`

Lower Pen: `Ec*p b`

Example: Lift the pen, move it to 100,100, draw a line to 200,200, then draw a second line to 50,300 (figure 3-2).

```
Ec*pa 100,100 200,200 50,300Z
```

Figure 3-2. Line Drawing Example



Defining Your Own Line Type

You can design your own line type for selection using the "`Ec*m <type> b`" sequence (described under the heading "Selecting the Line Type") It's done by representing the dot pattern of the line with a binary number, in which the meanings of the 1's and 0's are defined by the currently selected drawing mode (refer to "Drawing Modes" in Section 2).

A user defined line pattern is formed using two items: a dot pattern and a scale factor. The dot pattern is composed of a sequence of eight 1's and 0's. The scale factor indicates the number of times each bit in the dot pattern should be repeated to form the line pattern.

The dot pattern is specified as a decimal number from −32768 to 32767 from which a number from 0 to 255 is extracted by using the eight least significant bits of the 2's complement form of the number. For example, the dot pattern for the number 170 is `10101010`. The default dot pattern is `11111111`.

The scale factor indicates how many times each bit in the pattern is to be repeated to form the line pattern. For example, a scale factor of 3 applied to the dot pattern for the number 170 produces the following line pattern:

```
111000111000111000111000
```

This line pattern is used to form lines. The default scale factor is 1.

Assuming the default drawing mode, points indicated as a "1" in the line pattern are lighted on the screen; and points indicated as "0's" are unlighted.

The following escape sequence is used to define a line pattern:

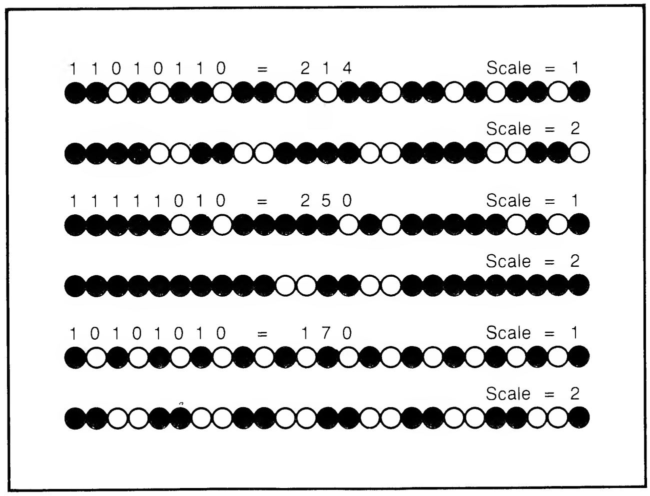
```
Ec*m <pattern> <scale> C
```

where:

<pattern>	is an decimal integer in the range −32768 through 32767. The line pattern is formed of the eight least significant bits of the binary equivalent of this integer. The default integer is 255 (all 1's).
<scale>	is a decimal integer in the range 1 through 255 which selects the number of times each bit of the pattern is to be repeated to form the expanded pattern, which is repeated as necessary to form the line. The default scale factor is 1.

Figure 3-3 shows examples of user defined line types.

Figure 3-3. Examples of User Defined Line Types



Example: Define a pattern to generate the following line:

`1111111111001100111111111001100`

Pattern	=	11111010 = 250
Scale	=	2
Escape sequence	=	Ec*m 250 2 C

Line patterns too complex to be obtained from an 8 X 8 area pattern, as described in "Filling Areas" can be generated by plotting a series of lines and varying the patterns used for successive lines. Complex patterns such as those used in weaving can be generated easily using this technique.

Using an Area Fill Pattern as a Line Type

By selecting line type 3 in the line type selection escape sequence, "`Ec*m <line type> b`", you can use an area fill pattern you have defined to draw horizontal and vertical lines. (Refer to "Filling Are1s" for information on area fill patterns.) Horizontal and vertical lines are drawn using the appropriate line from the area fill pattern. Diagonal lines cannot be drawn using the area fill pattern; solid lines are used instead. If the line is more than eight pixels long, the pattern is repeated to the end of the line.

Example: Plot three lines:

`(2,3)` to `(7,3)`
`(9,3)` to `(9,12)`
`(7,5)` to `(2,10)`

using a user defined area pattern of:

```
51,204,51,204,51,204,51,204
```

Figure 3-4 illustrates the area pattern and the lines drawn.

- Create the area fill pattern:

```
Ec*m 51,204,51,204,51,204,51,204D
```

- Select the user defined area fill pattern as the current area fill pattern:

```
Ec*m 2G
```

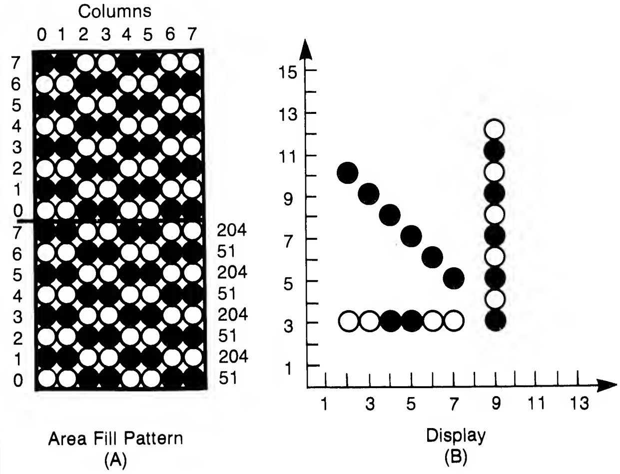
- Select the current area fill pattern as the current line type:

```
Ec*m3B
```

- Draw the lines:

```
Ec*p a 2,3 7,3 a 9,3 9,12 a 7,5 2,10 Z
```

Figure 3-4. Example of Lines Drawn Using a User Defined Area Pattern as a Line Type



Rubberband Line Mode

In Rubberband Line mode, the terminal displays a temporary line connecting the current pen position to the graphics cursor. As the cursor is moved, using any input device, the line moves, stretches, and contracts as required to maintain the connection. The temporary line is "set" (made permanent) when the cursor position is entered as a new point, by executing the "`Ec*pc`" command. The new point then becomes the current pen position and the process can be repeated.

On	`Ec*dm`
Off	`Ec*dn`

If the graphics cursor is not already on, entering Rubber-band Line mode automatically turns it on.

Specifying the Cursor Position as the Next Data Point

The escape sequence for using the cursor position as the next data point is:

```
Ec*pc
```

This sequence can be used with a tablet or a mouse to draw freehand lines.

Example: Position the cursor on the screen, type in "`Ec*pc`", and continue to hold down the [c] key while moving the cursor around the screen. A line will be drawn following the cursor.

Selective Erasing

A line drawn in Set mode can be selectively erased by redrawing it in Clear mode, although this will produce gaps if the erased line is intersected by other lines. This problem can be overcome by initially drawing the line in Complement mode, then redrawing it in Complement mode to erase the line. This technique will preserve the original display and can be useful for drawing and erasing temporary figures.

Example: Select Complement mode, draw a line, then erase the line by redrawing.

`Ec*m 3A`	Select Complement mode
`Ec*p a f 100,300 300,300Z`	Draw line
`Ec*p a f 100,300 300,300Z`	Erase line

Filling Areas

An area fill operation consists of selecting the area fill type and filling the area. When an area is specified for filling, the terminal fills the area with the area fill type currently selected.

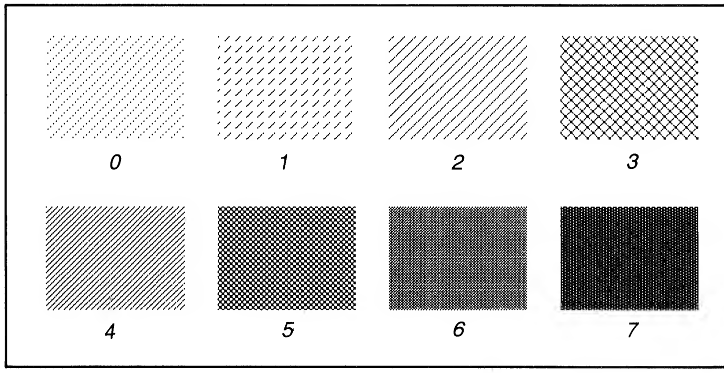
Selecting the Area Fill Type

Three types of area fill are available: solid, user defined, and predefined. Figure 3-5 illustrates the eight predefined area fill patterns. Use the following escape sequence to select the area fill type:

```
Ec*m <x>g
```

<x>	PATTERN
1	Solid area fill
2	User defined area fill (default)
3	Predefined pattern 0
4	Predefined pattern 1
5	Predefined pattern 2
6	Predefined pattern 3
7	Predefined pattern 4
8	Predefined pattern 5
9	Predefined pattern 6
10	Predefined pattern 7

Figure 3-5. Predefined Area Fill Patterns



Filling the Area

Two methods are available for specifying the area to be filled: rectangular and polygonal. Polygonal area specification is used for irregular areas.

Rectangular Area Fills. A rectangular area can be filled by sending the terminal the end points of a diagonal line in an escape sequence. The terminal fills the area, using the end points as corners of the rectangle, when it receives the escape sequence. The end point coordinates can be in either absolute or relocatable ASCII format.

Fill Rectangle, Absolute. This method of area specification is done with the following escape code:

```
Ec*m <x1><y1> <x2><y2> e
```

where `<x1><y1>` and `<x2><y2>` are the absolute coordinates of the two diagonal corners, respectively, of the fill area (−16384 to 16383). Spaces or commas can be used to separate the coordinates.

Example: Use predefined area fill pattern 5 to fill a rectangle defined by the coordinates 50, 50, 300, 300:

```
Ec*m 8 g 50,50 300,300 E
```

Fill Rectangle, Relocatable. Use the following escape sequence to specify a rectangle to be filled in relocatable coordinates:

```
Ec*m <x1><y1> <x2><y2>f
```

where `<x1><y1>` and `<x2><y2>` are the relocatable coordinates of the two diagonal corners, respectively, of the fill area (−32767 to 32767). Spaces or commas can be used to separate the coordinates.

Example: As a demonstration of this escape sequence, load a function key with the following string:

```
Ec*m 1 g 1 20,20 30,30 F
```

Then use the graphics cursor control keys to move the graphics cursor, and press the function key. Move the cursor and press the key several times.

Polygonal Area Fills. The terminal allows you to define a polygon composed of up to 148 sides, and fill it using the currently selected area fill method. The escape sequence, which includes the endpoint coordinates for each side of the polygon, is as follows:

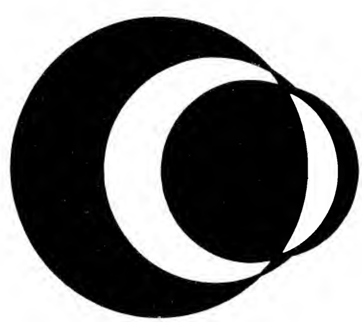
```
Ec*p <x> (coordinates)
```

where `<x>` is:

`<x>`	ACTION
a	Lift pen. Used to lift the pen so it can be moved to a new point. If a polygon is in process when the command occurs, it is closed.
s	Start a new polygon. This command causes subsequent coordinate pairs to be read as vertices of a new polygon.
t	Close the current polygon. When this command, or any capital letter, occurs in the escape sequence, the polygon is closed and filled using the currently selected area fill method.

Areas are filled in alternate order, as shown in figure 3-6.

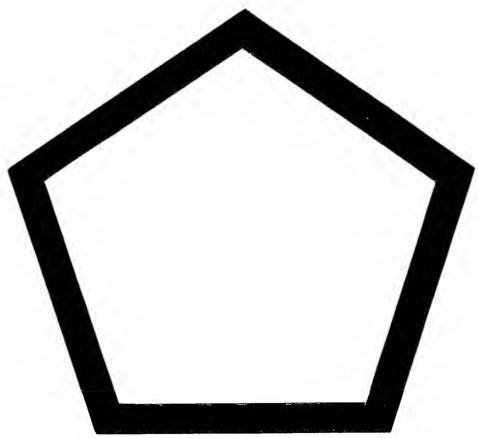
Figure 3-6. Area Fill of Overlapping Areas



Example: Move the pen to 33,0 and define and fill a pentagon 100 units on a side. Lift the pen, move it to 40,10, then define another pentagon inside the first one (figure 3-7).

```
Ec*p a s 33,0 133,0 166,95 83,150 0,95  a
40,10 12,91 83,138 153,91 125,10 T
```

Figure 3-7. Example of Polygonal Area Filling



Note

When using user-defined function keys to define polygonal area fills, the entire polygon specification must be contained in one function key, or some data may be lost.

Area Fill Patterns

When an area is specified for filling, the entire screen is divided into 8 X 8 cells. The content of each eight-pixel row of a cell is represented by a row pattern, with eight patterns required to define a cell pattern. To fill an area, the cell pattern is repeated for each cell in the area.

Defining Your Own Area Fill Pattern

A user defined area fill pattern is defined by specifying eight patterns in an escape code, one pattern for every row of pixels in the 8X8 cell. Each row pattern is specified in the escape code as a decimal number in the range −32768 through 32767. The terminal interprets the number as a 2's complement number, of which only the least significant eight bits are used, to obtain a value between 0 and 255. For the default drawing mode, 1 bits are drawn and 0 bits are not drawn. Other drawing modes interpret the bits differently. (Refer to "Drawing Modes", in this section, for more information on drawing modes.)

The escape sequence for defining an area fill pattern is as follows:

```
Ec*m <row 0> <row 1> .....<row 7>d
```

where the `<row>` parameters are decimal numbers which specify the patterns for the eight rows of the cell.

Example: Define a simple checkerboard pattern for an 8 X 8 cell.

ROW	BINARY	DECIMAL
0	10101010	170
1	01010101	85
2	10101010	170
3	01010101	85
4	10101010	170
5	01010101	85
6	10101010	170
7	01010101	85

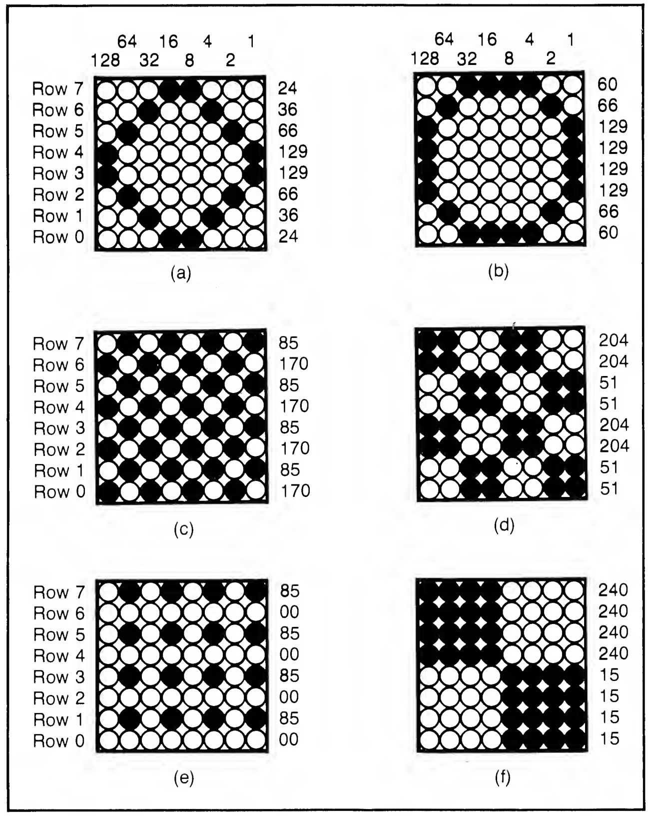
```
Ec*m170 85 170 85 170 85 170 85D
```

Note

The scale factor of an area fill pattern is always 1.

Figure 3-8 illustrates examples of some user defined area fill patterns.

Figure 3-8. Examples of User Defined Area Fill Patterns



Using Graphics Text

Text strings can be written directly into graphics memory. An internal character generator converts the ASCII codes into a vector list representation which is then drawn as lines.

The character set includes upper and lower case (95 characters) and the national characters shown in Appendix D. The national character sets are selected on the Terminal Configuration menu, discussed in Section 6. The characters are drawn in a 5 X 7 matrix in a 7 X 10 cell, with descenders for lower case.

This character set is in addition to the normal alphanumeric character set. While this character set may seem redundant, it offers the following advantages:

- Characters can be drawn at any dot position, rather than 24 by 80 alphanumeric character positions.
- Characters can be rotated in multiples of 90 degrees.
- Characters can be scaled in size, from 1 to 8 times.
- Characters can be slanted 27 degrees for an italics-like effect.
- Lines of characters can be right, left, or center justified.

Figure 3-9 shows the graphics character set.

Figure 3-9. Graphics Text Character Set

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ê ô û á é ó ú à è ò ù ä ë ö ü Å î Ø Æ å í ø æ Ä ì Ö

! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 :

a ´ ` ^ ¨ ˜ £ ¯ ˇ ç Ñ ñ ¡ ¿ ¤

; < = > ? @ [\ ] ^ _ ` { ¦ } ~

£ § â ü É ï ß

Foreign Characters

All of the USASCII and international character sets are available by entering "Yes" in the "ASCII 8 Bits" field of the Terminal Configuration menu, then entering Extended Characters mode by pressing [CTRL] and [.], simultaneously. You leave Extended Characters mode by simultaneously pressing [CTRL] and [,]. You can also access the extended characters by pressing the [Extend char] key, together with the appropriate keyboard key.

The normal sequence of operations for using graphics text is:

- Select the desired characteristics for the text. The selectable characteristics are: size, orientation, slant, and location.
- Turn on Graphics Text mode.
- Enter the text.
- Turn off Graphics Text mode.

The currently-selected or default characteristics can be used, if desired. The last-selected characteristics remain the currently-selected ones until they are changed. After a power on or hard reset, the default characteristics become active. Refer to "Selecting the Graphics Default Parameters", later in this section, for a list of default parameters.

From the Keyboard

Graphics text can be entered directly from the keyboard. To select the text characteristics and turn on and off Graphics Text mode, the escape codes described in the following paragraphs are entered at the keyboard.

Tabbing right spaces one graphics text character to the right. Tabbing vertically ([CTRL] and [K], pressed together) spaces one graphics text line up. (The actual direction of movement will depend on the text orientation.)

The backspace, carriage return, and line feed functions work as expected (even on inverted text), making it easy to add or edit titles and labels. In addition, the following keys function in the same manner as for alphanumeric text characters:

[Return], [Back space], [◄], [►], [▲], [▼].

From a Program

All graphics text characteristics can be selected programmatically using the escape codes described in succeeding paragraphs.

Selecting Text Characteristics

The selectable text characteristics are size, orientation, slant, and location on the screen.

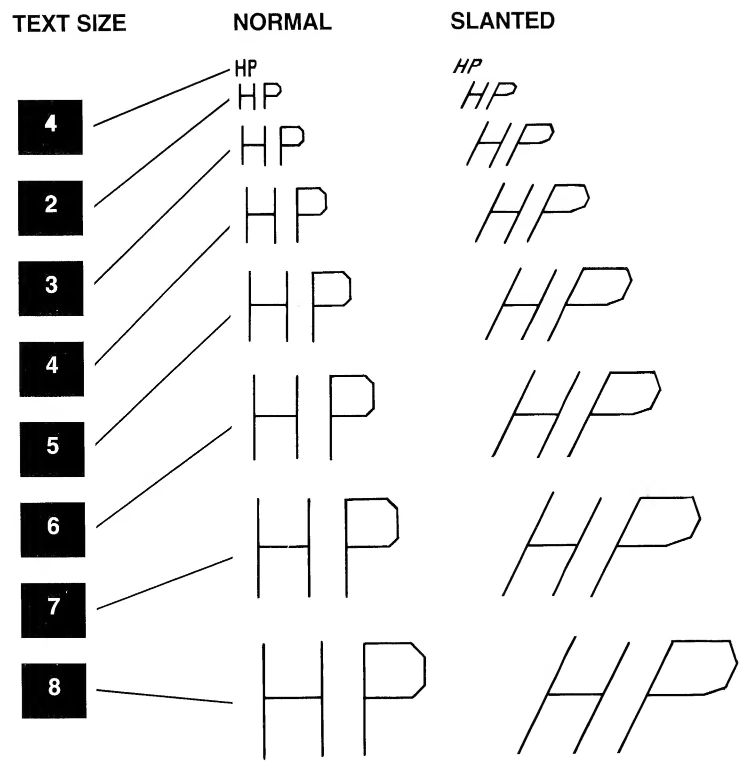
Size. The ASCII characters 1 through 8 specify the character size for graphics text. A "1" indicates the smallest character, a 5 X 7 dot matrix in a 7 X 10 cell. Increasing the size increases the length of the vectors. If a text size of 1 is specified, each vector in the list is drawn on the screen unscaled. A size of 2 results in vectors which are twice as long, etc. (figure 3-10). A size of "1" is the default.

Use the following escape sequence to set the graphics text size:

```
Ec*m <size> m
```

where `<size>` is a number from 1 to 8.

Figure 3-10. Graphics Text Sizes



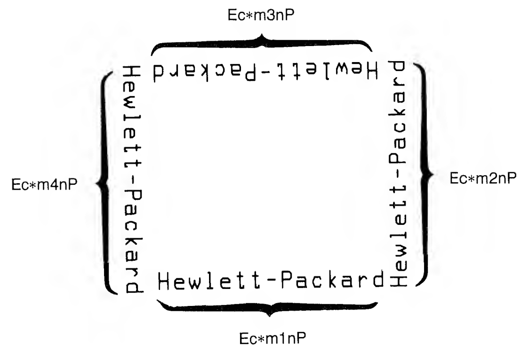
Orientation. This command uses the ASCII characters 1 through 4 to specify the text orientation (figure 3-11). It also changes the direction of line feed, carriage return, and backspace. The following escape sequence sets the text orientation:

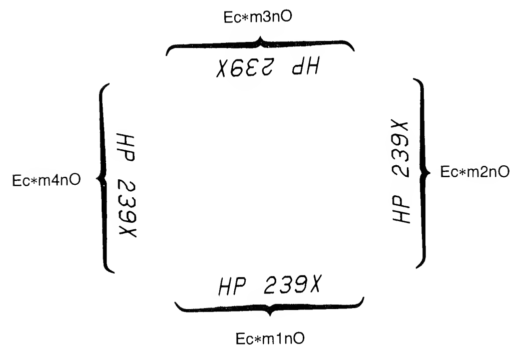
```
Ec*m <x> n
```

where:

<x>	ORIENTATION
1	Normal (upright, the default)
2	Rotated 90 degrees counter clockwise
3	Rotated 180 degrees counter clockwise (inverted)
4	Rotated 270 degrees counter clockwise

Figure 3-11. Graphic Text Orientations and Slant





Slant. The graphics text characters can be slanted 27 degrees for an italics effect. The default selection is no slant. Figure 3-11 illustrates slanted text.

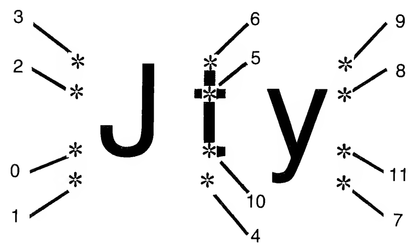
ON	`Ec*m o`
OFF	`Ec*m p`

Locating the Text on the Screen. Text strings can be right or left justified, or centered about the current cursor position, which serves as a reference point.

To position text on the screen:

- Select one of 12 origin points on the text string (figure 3-12). When the text is entered on the screen, the origin will overlay the cursor. The default origin is 1.
- Position the cursor so that the text string will be located in the desired position.

Figure 3-12. Selectable Origins for Graphics Text Location



You select the origin with the following escape sequence:

```
Ec*m <origin> q
```

where `<origin>` can be any number in the range 0–11.

If the origin is selected as 0, 1, 2, or 3, the text begins at the cursor position. If 4, 10, 5, or 6 is selected as the origin, the text is centered, horizontally, on the cursor position. Selecting 7, 11, 8, or 9 locates the text to end at the cursor position.

Bottom of text cell (1, 4, and 7), middle of text cell (2, 5, and 8), top of text cell (3, 6, and 9), and text baseline (0, 10, and 11) origin points select the reference line for the line of text. For example, if text is to be right justified and the base line set at the top of the character position, the number "9" should be selected as the origin.

If left justification is selected, the pen position moves to the cursor position as soon as text is entered.

When centering or right justification is used, the text strings are buffered (stored) until all of the characters in the string have been received. The string is not displayed until the `CR` or `LF` is received, which can be confusing when entering text from the keyboard. The string end is detected by a `CR` or `LF`.

The maximum length of a center- or right-justified string is 73 characters (not including the `CR` (`LF`)). In all cases, data written beyond the edge of the screen is lost. There is no automatic `CR` when the screen boundary is reached.

Turning Graphics Text Mode On and Off

Graphics text mode is turned on and off with the following escape sequences:

ON	`Ec*d S`
OFF	`Ec*d T`

In Graphics Text mode, all displayable characters are stored in graphics memory. The drawing mode is initially set to Jam1 mode to permit overstrike replacement of characters. A different drawing mode, such as Set or Complement, can be selected at any time.

Text is drawn using the current text assignments for size and orientation. Graphics text mode accepts `CR`, `LF`, `BS`, `HT`, and `VT` as control characters. The ◄, ►, ▲, ▼ keys can be used to position the graphics cursor in character increments.

If the graphics cursor is moved, the graphics text margin is moved to the new cursor or pen position.

Entering Text

Two methods are available for entering graphics text: normal text entry and entering a label. Both use the currently selected text characteristics. For normal text entry, the text string is located with reference to the current graphics cursor position. A text string entered as a label is located with reference to the current pen position, which may not be the same as the cursor position.

Turning on Rubberband Line mode will show if the cursor and pen position are at different locations. Since the rubberband line stretches between the current pen position and the cursor, no rubberband line will be present if the cursor overlies the pen position.

Normal Text Entry. For normal text entry, graphics text can be entered immediately after turning on Graphics Text mode. When entered programmatically, the text string can be included in the escape sequence which turns on Graphics Text mode.

Upon entering Graphics Text mode, the pen position moves to the current cursor position. When the text string is ended, with a carriage return, the cursor and pen position return to the left margin of graphics text, which is located at the leftmost entered character.

Note

As text is entered in Graphics Text mode, the pen position tracks the cursor position, so that, even though Rubberband Line mode is on, no rubberband line will show.

Example: Turn on Graphics Text mode, turn on the graphics cursor, position the cursor at 100,100, and
enter, as graphics text, the text string
"`This a text string`":

```
Ec*d s k 100,100 O This is a text string
```

Labels. Labels are used to send a single record of graphics text to the terminal without entering Graphics Text mode. The characters are stored in graphics memory, using the current text size, angle, slant, and justification, until a CR or LF is received; then they are entered on the screen.

The record must end with a CR, LF, or both. A CR moves the pen to its original position when the label command was received. An LF moves the pen down one line (character spacing). The actual directions moved following a CR or LF depend on the text orientation selected.

The differences between normal text entry and labels are:

- Labels can be entered without entering Graphics Text mode.
- Only 73 text characters or less can be entered as a label. The label is ended by a carriage return.
- The label reference point, which is overlaid by the selected justification origin to locate the text, is the current pen position. For normal text entry, it is the current cursor position.
- The graphics cursor does not change position when a label is entered. Only the pen position moves.

Labels are entered with the following escape sequence:

```
Ec*l <text string> CR(LF)
```

Example: Enter, from the keyboard, the label text string "`This is a sample label`":

```
Ec*l This is a sample label <CR>
```

where `<CR>` is a carriage return produced by pressing the [Return] key.

Graphics Text Status

You can check the current text settings with a graphics text status request. Refer to "Graphics Status" in Section 9 for additional information.

Obtaining a Hardcopy

You can obtain a hardcopy of the screen display using either a printer or plotter. However, a plotter copy can only be obtained through use of a program, such as HPDRAW, which stores the picture data as it is entered at the terminal and transfers it and the necessary commands to the plotter.

With a suitable printer, capable of accepting raster dump data, connected to the external device port, you can obtain a copy of the contents of raster memory using the keyboard or a program which requests a raster dump.

You control the size, layout, and content of the copy by configuring the appropriate configuration menu. The External Parallel Device Configuration menu is for parallel-input devices, and the External HP-IB Device Configuration menu is used for HP-IB-connected devices. Refer to Section 7 for copying information.

Using a Relocatable Origin

Using relocatable origins is a convenient way to draw a given figure at a number of screen locations. Since the figure is specified in terms of coordinates, it is much easier, for each drawing of the figure, to specify a new origin to which the coordinates of the figure are referenced than to generate a new set of coordinates.

To be referenced to a relocatable origin, the figure coordinates must be in relocatable format, either ASCII or binary.

When a relocatable origin is specified, the figure is located on the screen by adding the X coordinate of the relocatable origin to each of the X coordinates of the drawing. This is also done for the Y coordinates.

A relocatable origin can be referenced to one of three points: the absolute origin, the current pen position, or the current cursor position.

Set Relocatable Origin, Absolute

The relocatable origin can be set to any absolute coordinates using ASCII absolute format (−16384 to 16383). The escape sequence is as follows:

```
Ec*m <X,Y> j
```

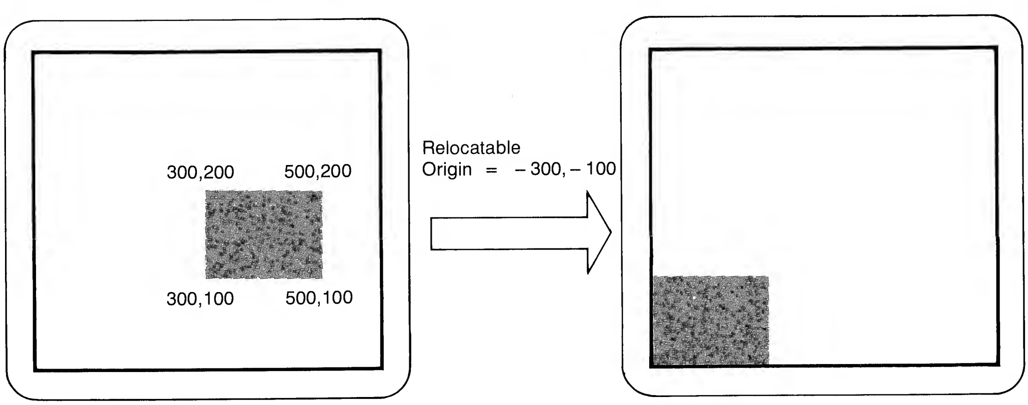
where: `<X,Y>` are the x and y coordinates of the new origin in ASCII absolute format.

Example: Set the relocatable origin to display the box in the figure so that the box is positioned at the lower left corner of the display (figure 3-13).

`Ec*m -300,-100 J`	Set new origin

`Ec*p a h 300,100 500,100`
`500,200 300,200`
`300,100 Z`
	Draw figure

Figure 3-13. Relocatable Origin Use



Set Relocatable Origin to Current Pen Position

The relocatable origin can be set to the current pen position using either of two escape sequences:

```
Ec*p e
```

or

```
Ec*m k
```

Set Relocatable Origin to Graphics Cursor Position

The relocatable origin can be set to the current graphics cursor position.

```
Ec*m l
```

Graphics Status

An escape sequence is available for requesting various kinds of graphics status information, such as pen and cursor positions, location of the relocatable origin, size, slant, angle, and origin of graphics text, etc. Refer to "Graphics Status" in Section 9, Status, for detailed information.

Selecting the Graphics Default Parameters

Graphics parameters can be set to their default (power on or full reset) values. Table 3-1 lists the various parameters and their default values. Additional information can be found under the discussions of individual parameters. The escape sequence for selecting the graphics default values is:

```
Ec*m r
```

The current graphics mode and settings can be obtained with graphics status requests. Graphics status requests are described in the Status section. It may be desirable to reselect graphics settings before you send graphics data to the terminal.

Table 3-1. Graphics Parameter Default Values

PARAMETER	DEFAULT VALUE
*Pen Condition	Down
*Line Type	1 (solid)
*Drawing Mode	2 (JAM 1)
*User Defined Line Pattern	255, 1
*Area Fill Type	2 (User Defined Pattern)
*User Defined Area Fill Pattern	255, 255,..., (Solid)
*Background Pen	0 (Black)
*Primary Pen	7 (White)
*Secondary Pen	0 (Black)
*Boundary Pen	Off
*Graphics Text	Off
*Text Size	1
*Text Direction	1
*Text Origin	1 (left, bottom)
*Text Slant	Off
*Text Color	Primary Pen
Relocatable Origin	0,0
Alpha Video	On
Graphics Video	On
Alpha Cursor	On
Graphics Cursor	Off
Graphics Cursor Address	0,0
Rubberband Line	Off
Compatibility Mode:	
Page Full Straps	0 (Out)
GIN Strap	0 (CR only)

Display Functions Mode

The DISPLAY FUNCTNS key, in the Modes set of function keys, is used to activate and deactivate Display Functions mode.

When Display Functions mode is inactive (normal operation), graphics commands received by the terminal are executed when received. These commands might be graphics keypad keystrokes or escape sequences.

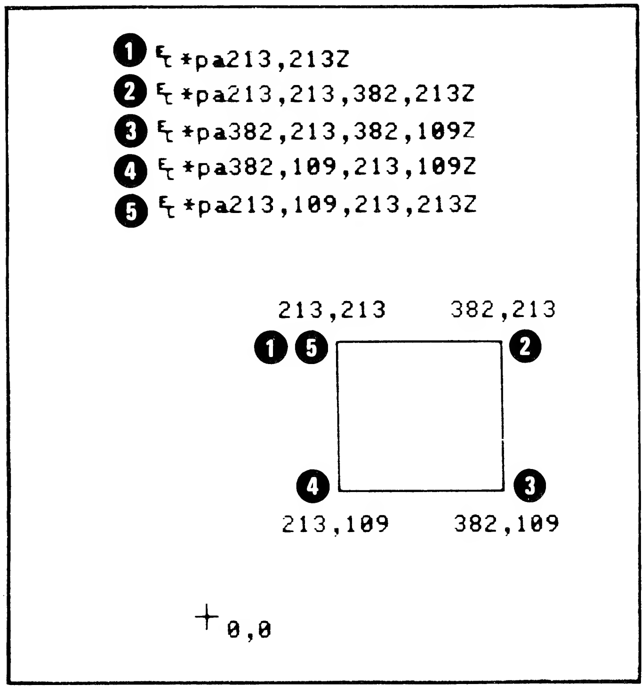
When Display Functions mode is active, escape sequences, instead of being executed, are stored in alphanumeric display memory and displayed on the screen. Table 3-2 lists the graphics keypad keys and associated escape sequences which operate as described above in Display Functions mode.

Table 3-2. Escape Sequences Associated with Graphics Keys

KEY	SEQUENCE	DESCRIPTION
◀ ▲ ▶ ▼	None	Graphics cursor control keys
Cursor fast	None	Increases speed of graphics cursor
Graph clear	Ec*d A	Clear graphics memory
Graph display	Ec*d C Ec*d D	Turn graphics display on Turn graphics display off
Alpha display	Ec*d E Ec*d F	Turn alpha display on Turn alpha display off
Graph cursor	Ec*d K Ec*d L	Turn graphics cursor on Turn graphics cursor off
RB Line	Ec*d M Ec*d N	Turn rubberband line on Turn rubberband line off
Graph device	None	Turns HP-HIL device off
Draw	Ec*p bC	Draws a line from the current pen position to the cursor position
Move	Ec*p aC	Moves pen position to cursor position

Figure 3-14 shows the sequences generated when drawing a simple box. The graphics cursor is initially on and positioned at 0,0.

Figure 3-14. Displaying Graphics Escape Sequences



Compatibility Mode

Compatibility mode allows the terminal to plot data intended for 4010 or 4014 terminals with displays of 1024 X 1024 or 4096 X 4096 addressable points. Thus, with a minimum of reprogramming, programs developed for use with 4010 and 4014 terminals can be used with this terminal.

Lines are drawn using the current line type and line drawing mode. This gives you the capability of drawing dotted and dashed lines, etc. by changing the program to send the additional escape sequences.

Handshaking

When not in Compatibility mode, the terminal will normally respond to an "Enq" character with an "Ack" character. In Compatibility mode, the Enq/Ack handshake is disabled; most control characters (ASCII decimal codes 0-31) are ignored. However, XON/XOFF handshaking can be used.

Control and Escape Sequence Variations

Certain control and escape sequences, in Compatibility mode, have different meanings or are ignored in HP mode. These sequences and their Compatibility mode actions are listed in table 3-3.

Table 3-3. Control and Escape Sequences which Differ Between HP and Compatibility Modes

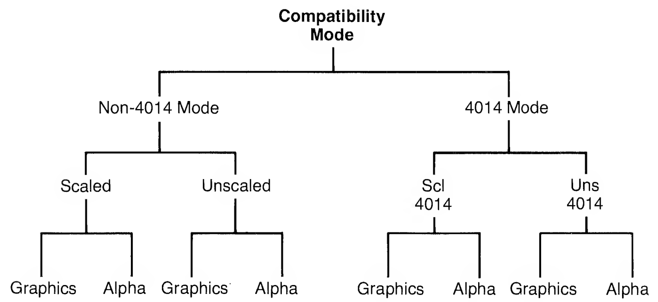
CODE	ASCII DECIMAL CODE	KEYBOARD CHAR	OPERATION
BS	8	CTRL H	Backspace. Moves one space left.
HT	9	CTRL I	Horizontal tab. Moves one space left.
LF	10	CTRL J	Line feed. Moves one line down.
VT	11	CTRL K	Vertical tab. Moves one line up.
CR	13	CTRL M	End Graphics mode.
GS	29	CTRL]	Enter Graphics mode.
US	31	(none)	Enter Alphanumeric mode.
Ec FF	27 12	ESC L	End Graphics mode, clear screen, and home cursor.
Ec EB	27 23	ESC W	Make hardcopy.
*Ec EQ	27 5	ESC E	Read status and alphanumeric cursor position.
*Ec SB	27 26	ESC Z	Read graphics cursor position.

*** Note:** Refer to the Status section for information on these sequences.

Submodes

Compatibility mode has several submodes, as shown in figure 3-15.

Figure 3-15. Submodes of Compatibility Mode



The number of addressable pixels for each mode, and the terminal to which each mode is related are listed below (EGM is an abbreviation for Enhanced Graphics Module).

TERMINAL	MODE	ADDRESSABLE PIXELS
HP	Low resolution	512 X 390
HP	High resolution	640 X 400
4010	Scaled or Unscaled	1024 X 780
4014 without EGM	Scaled or Unscaled	1024 X 780
4014 with EGM	Scl 4014 or Uns 4014	4096 X 3120

Non-4014 Mode. Non-4014 mode is for emulation of 4010 terminals and 4014 terminals without the Enhanced Graphics Module (EGM). (The EGM increases the resolution of 4014 terminals from 1024 X 1024 to 4096 X 4096).

4014 Mode. 4014 mode is reserved for emulation of 4014 terminals with the EGM installed. In 4014 mode the screen resolution is extended to 4096 X 4096 addressable pixels (4096 X 3120 displayable). Two binary bits of precision are added to the address. Pixels are addressed as 0.25, 0.5, 0.75, 1.0, etc.

The two added binary bits are specified by an "extra byte", allowing 4010 graphics, which use only the integer addresses of the 4014, to appear full-sized on the 4014 mode display. Each coordinate of an address is divided by 8 to fit the entire picture on the 512 by 390 screen. Graphics drawn in this mode have the same appearance on the screen as graphics drawn in Scaled mode.

Variable Character Sizes. In 4014 mode, the terminal ignores commands for changing character size. Thus the sequences "`Ec 8`", "`Ec 9`", "`Ec :`", and "`Ec ;`" are not executed in 4014 mode.

8-Bit Mode. Characters sent to the terminal in 4014 mode have their parity bit cleared automatically. Thus the Roman Extension Character Set is not accessible in 4014 mode. Seven-bit mute processing retains its normal function.

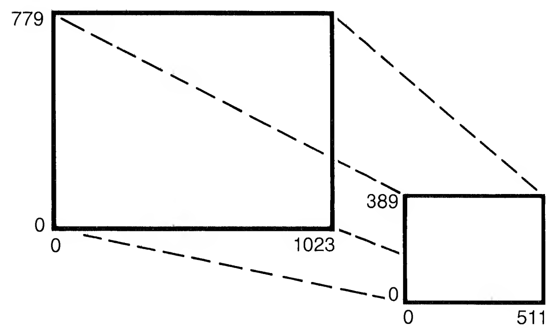
Incremental Point Plot. Sending the terminal an "`RS`" control character, in 4014 mode, initiates incremental point plot operation. Commands for pen up (SP)(i.e., an ASCII "Space") or pen down (`P`) control line drawing in this mode. The terminal draws a single dot as the graphics beam moves in one-point increments according to the following directional commands: `D`—North; `E`—Northeast; `A`—East; `I`—Southeast; `H`—South; `J`—Southwest; `B`—West; `F`—Northwest.

Point Plot. An "FS" control code selects point plot mode operation. As for HP 4010 line type 11, only the last point of a line is drawn.

Special Point Plot Mode. The terminal displays lines drawn in this mode, but does not vary the intensity of the graphics beam.

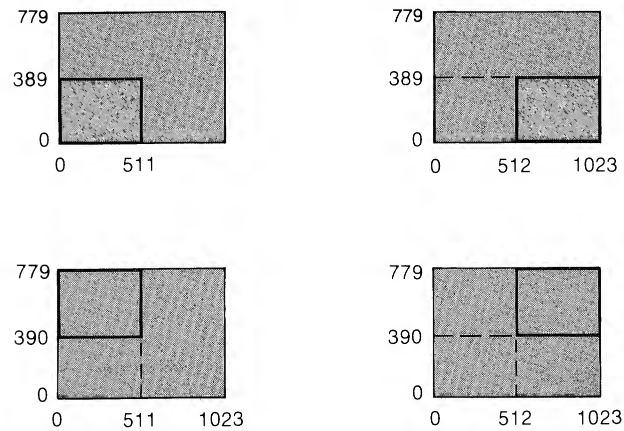
Scaled Mode. Scaled mode reduces the larger displays of the 4010 terminal and the 4014 terminal without the EGM to the smaller HP screen size.This is done by dividing the x and y coordinates by 2, mapping the 1024 X 780 pixel display to 512 X 390 pixels (figure 3-16).

Figure 3-16. Image Area Scaling in Scaled Mode



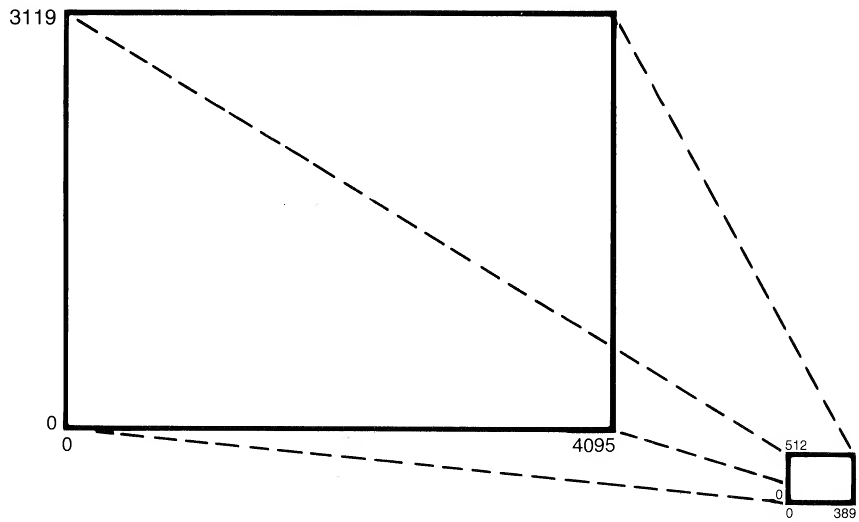
Unscaled Mode. Unscaled mode allows display, on the HP screen, of unreduced portions of the larger display of the 4010 terminal and the 4014 without the EGM. A relocatable origin is selected such that a 512 X 390 pixel part of the larger display is shown on the screen. The entire picture can be displayed, in increments, by placing the relocatable origin so that the displayed part of the picture adjoins the previously displayed part (figure 3-17).

Figure 3-17. Using Unscaled Mode and a Relocatable Origin to Display a 1024 X 780 Image Area on a 512 X 390 Screen



SCL 4014 Mode. Scl 4014 mode reduces the 4096 X 3120 pixel display of the 4014 terminal with the EGM installed to the 512 X 390 HP display size. It does so by dividing each x and y coordinate by 8 (figure 3-18).

Figure 3-18. Image Area Scaling in Scl 4014 Mode



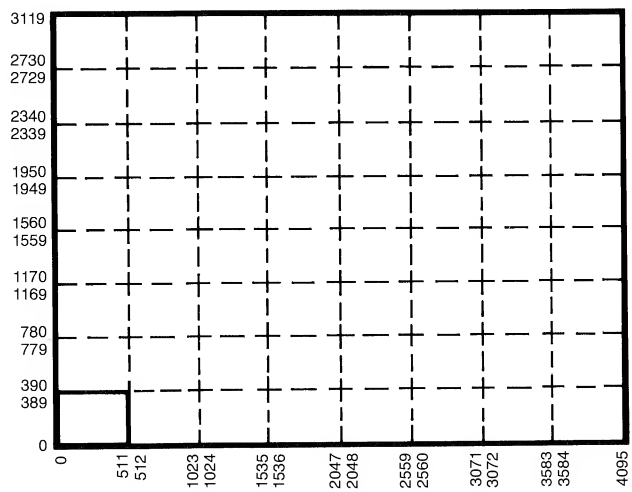
Whereas the 4010 draws only solid lines, the 4014 has five line types: solid, dotted, short-dash, long-dash, and dot-dash. Escape sequences for these line types are accepted in SCL 4014 mode. These sequences retain their HP definitions in Scaled mode. The following commands select the line type for line drawing:

`Ec '`	for normal lines or alphanumeric data
`Ec a`	for dotted line lines
`Ec b`	for dot-dashed lines
`Ec c`	for short-dashed lines
`Ec d`	for long-dashed lines

The sequences `Ec <h through l>` and `Ec <p through t>` can also be used to select line types. The terminal accepts the line type specified by the sequence but ignores the 4014 line-width definition (defocused and write-thru lines are not supported on the HP terminal.)

UNS 4014 Mode. Uns 4014 mode is used to display unreduced 512 X 390 portions of the 4096 X 3120 display of the 4014 terminal with the EGM installed. Using a relocatable origin, the entire image area can be displayed, in sections (figure 3-19).

Figure 3-19. Using Uns 4014 Mode and a Relocatable Origin to Display a 4096 X 3120 Image Area on a 512 X 390 Screen



Alphanumeric and Graphics Modes. Alphanumeric and Graphics modes are for entering data into either alphanumeric or graphics memory. Normally, the terminal is switched between these two modes to display messages, plot graphics figures, and then display additional messages. The default selection, which occurs when Compatibility mode is entered, is Graphics mode.

In general, all the normal features of the terminal (display enhancements, etc.) are available only in Alphanumeric mode.

Selecting Modes

From the Keyboard. Compatibility mode is turned on by selection, in the "`Graph Compat`" field of the Global Configuration menu, of any one of the four submodes: Scaled, Unscaled, Scl 4014, and Uns 4014. No selection is required for 4014 mode. To switch between Graphics and Alphanumeric modes, the escape codes (covered below) must be used.

To return to normal HP operation, Compatibility mode is turned off by selection of "`Off`" in this field.

From a Program. From a program, Compatibility mode is initiated by selecting either Scaled or Unscaled mode directly or selecting 4014 mode, then selecting either Scl 4014, or Uns 4014 mode. Normal HP mode is restored by turning off Compatibility mode. Graphics and Alphanumeric modes are selected while in one of the above-listed modes.

4014 Mode. The following escape sequence enables control of 4014 mode.

```
Ec*t <x>D
```

where:

<x>	ACTION
0	4014 mode off (default).
1	4014 mode on.

Submode Selection. With 4014 mode initiated, you can use the following escape sequence to select one of four submodes:

```
Ec&s <x>p <y>Q
```

where:

<x>	<y>	4014 MODE	ACTION
0	1	Off	Turn on Unscaled mode.
0	1	On	Turn on Uns 4014 mode.
1	0	Off	Turn on Scaled mode.
1	0	On	Turn on Scl 4014 mode.

Graphics and Alphanumeric Modes. After selecting either Scaled, Unscaled, Scl 4014, or Uns 4014, you can switch between Graphics and Alphanumeric modes using certain ASCII characters: group separator (GS), form feed (FF), unit separator (US), escape character (Ec), and carriage return (CR).

CHAR	ACTION
GS	Enter Graphics mode.
CR	End Graphics mode.
Ec FF	End Graphics mode, clear screen, and home cursor.
US	Enter Alphanumeric mode.

Ending Compatibility Mode

You can return to the original operating mode (previous to entering Compatibility mode), from any submode, with the following escape sequence:

```
Ec&s 0p 0Q
```

Configuration Selections

When in Compatibility mode, you can select the following capabilities, derived from the terminal being emulated. Refer to the manual for the terminal being emulated for additional information on the operation of these straps and how they should be set.

Graphic Input Terminator. You can select the terminator sent by the terminal following the input of cursor address information. The terminator can be a CR, CR and EOT, or no terminator.

Page Full Busy. When this strap is in, the keyboard will be locked after the 35th line of text is received from the computer. The terminal can be cleared by pressing the [Graph clear] key. This strap is ignored in Unscaled mode.

Page Full Break. When this strap is in, the terminal will send a break signal (selectable on either of the datacomm menus) to the computer after the 35th line of text is displayed. The terminal may also be set to BUSY (see Page Full Busy). When out, the strap will cause the cursor to home and the next 35 lines of text to be set with a left margin at X = 259. This strap is ignored in Unscaled mode.

Graphics Data Format

Depending on whether or not the terminal being emulated has the EGM installed, either four or five bytes are used to represent coordinate addresses.

4010 Terminals and 4014 Terminals without the EGM. Four bytes are required to represent the coordinates of a point for 4010 terminals and 4014 terminals without the EGM installed (0–1023). The contents of each byte is shown below:

BYTE	BIT						
	7	6	5	4	3	2	1
HI Y	0	1	Y10	Y9	Y8	Y7	Y6
LOW Y	1	1	Y5	Y4	Y3	Y2	Y1
HI X	0	1	X10	X9	X8	X7	X6
LOW X	1	0	X5	X4	X3	X2	X1

For emulation of terminals without the EGM, graphics data sent to the terminal must have the "Y" coordinate sent first:

```
<Hi Y> <Low Y> <Hi X> <Low X>.
```

When data is returned to the computer (cursor position, etc.), the X coordinate is returned first:

```
<Hi X> <Low X> <Hi Y> <Low Y>.
```

4014 Terminals with the EGM. The greater number of addressable pixels on terminals with the EGM installed requires five bytes to address all addressable pixels (0–4095). The content of each byte is listed below:

BYTE	BIT						
	7	6	5	4	3	2	1
HI Y	0	1	Y12	Y11	Y10	Y9	Y8
EXTRA BYTE	1	1	*	Y2	Y1	X2	X1
MID Y	1	1	Y7	Y6	Y5	Y4	Y3
HI X	0	1	X12	X11	X10	X9	X8
MID X	1	0	X7	X6	X5	X4	X3

* Bit 5 of the Extra byte can be used to set Margin 1.

For terminals with the EGM, graphics data sent to the terminal must be sent in the following order:

```
<Hi Y> <Extra byte> <Low Y> <Hi X> <Low X>.
```

For data transmitted from the terminal to the computer, the following order is required:

```
<Hi X> <Low X> <Hi Y> <Extra byte> <Low Y>.
```

Abbreviated Addressing. Data bytes sent to the terminal use bits 6 and 7 to indicate the byte is a Hi byte, a low Y, a low X or an Extra byte. Bit 8 (parity) is not used.

7	6	
0	1	Hi X or Hi Y byte
1	0	Low X byte
1	1	Low Y byte (terminals without the EGM)
1	1	Low Y, Low X byte (terminals with the EGM)

These identifying bits allow you, in many cases, to send less than the full four or five bytes when an address changes.

For the terminals using a four byte address, the following data bytes must always be sent:

- Low X byte.
- Any changed byte.
- Low Y byte, if the Hi X byte has changed.

For terminals using a five byte address, the preceding rules apply, with the following addition. When the Extra byte changes, send the following:

- Extra byte.
- Low Y byte.
- Low X byte.

Address Codes. ASCII characters are used as code, in place of numbers, for the point coordinates. Table 3-4 can be used to determine the ASCII characters to be used in the address bytes.

Table 3-4. ASCII Characters Used as Code for Addressing Point Coordinates

X or Y Coordinate																Low Order Y DEC.	Low Order Y ASCII	Low Order X DEC.	Low Order X ASCII
0	32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	96	`	64	@
1	33	65	97	129	161	193	225	257	289	321	353	385	417	449	481	97	a	65	A
2	34	66	98	130	162	194	226	258	290	322	354	386	418	450	482	98	b	66	B
3	35	67	99	131	163	195	227	259	291	323	355	387	419	451	483	99	c	67	C
4	36	68	100	132	164	196	228	260	292	324	356	388	420	452	484	100	d	68	D
5	37	69	101	133	165	197	229	261	293	325	357	389	421	453	485	101	e	69	E
6	38	70	102	134	166	198	230	262	294	326	358	390	422	454	486	102	f	70	F
7	39	71	103	135	167	199	231	263	295	327	349	391	423	455	487	103	g	71	G
8	40	72	104	136	168	200	232	264	296	328	360	392	424	456	488	104	h	72	H
9	41	73	105	137	169	201	233	265	297	329	361	393	425	457	489	105	i	73	I
10	42	74	106	138	170	202	234	266	298	330	362	394	426	458	490	106	j	74	J
11	43	75	107	139	171	203	235	267	299	331	363	395	427	459	491	107	k	75	K
12	44	76	108	140	172	204	236	268	300	332	364	396	428	460	492	108	l	76	L
13	45	77	109	141	173	205	237	269	301	333	365	397	429	461	493	109	m	77	M
14	46	78	110	142	174	206	238	270	302	334	366	398	430	462	494	110	n	78	N
15	47	79	111	143	175	207	239	271	303	335	367	399	431	463	495	111	o	79	O
16	48	80	112	144	176	208	240	272	304	336	368	400	432	464	496	112	p	80	P
17	49	81	113	145	177	209	241	273	305	337	369	401	433	465	497	113	q	81	Q
18	50	82	114	146	178	210	242	274	306	338	370	402	434	466	498	114	r	82	R
19	51	83	115	147	179	211	243	275	307	339	371	403	435	467	499	115	s	83	S
20	52	84	116	148	180	212	244	276	308	340	372	404	436	468	500	116	t	84	T
21	53	85	117	149	181	213	245	277	309	341	373	405	437	469	501	117	u	85	U
22	54	86	118	150	182	214	246	278	310	342	374	406	438	470	502	118	v	86	V
23	55	87	119	151	183	215	247	279	311	343	375	407	439	471	503	119	w	87	W
24	56	88	120	152	184	216	248	280	312	344	376	408	440	472	504	120	x	88	X
25	57	89	121	153	185	217	249	281	313	345	377	409	441	473	505	121	y	89	Y
26	58	90	122	154	186	218	250	282	314	346	378	410	442	474	506	122	z	90	Z
27	59	91	123	155	187	219	251	283	315	347	379	411	443	475	507	123	{	91	[
28	60	92	124	156	188	220	252	284	316	348	380	412	444	476	508	124	¦	92	\
29	61	93	125	157	189	221	253	285	317	349	381	413	445	477	509	125	}	93	]
30	62	94	126	158	190	222	254	286	318	350	382	414	446	478	510	126	~	94	^
31	63	95	127	159	191	223	255	287	319	351	383	415	447	479	511	127	■	95	_
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	← DEC.			
SP	!	"	#	$	%	&	'	(	)	*	+	,	–	.	/	← ASCII			
High Order X & Y																			

Table 3-4. ASCII Characters Used as Code for Addressing Point Coordinates (continued)

X or Y Coordinate																Low Order Y		Low Order X	
																DEC.	ASCII	DEC.	ASCII
512	544	576	608	640	672	704	736	768	800	832	864	896	928	960	992	96	`	64	@
513	545	577	609	641	673	705	737	769	801	833	865	897	929	961	993	97	a	65	A
514	546	578	610	642	674	706	738	770	802	834	866	898	930	962	994	98	b	66	B
515	547	579	611	643	675	707	739	771	803	835	867	899	931	963	995	99	c	67	C
516	548	580	612	644	676	708	740	772	804	836	868	900	932	964	996	100	d	68	D
517	549	581	613	645	677	709	741	773	805	837	889	901	933	965	997	101	e	69	E
518	550	582	614	646	678	710	742	774	806	838	870	902	934	966	998	102	f	70	F
519	551	583	615	647	879	711	743	775	807	839	871	903	935	967	999	103	g	71	G
520	552	584	616	648	680	712	744	776	808	840	872	904	936	968	1000	104	h	72	H
521	553	585	617	649	681	713	745	777	809	841	873	905	937	969	1001	105	i	73	I
522	554	586	618	650	682	714	746	778	810	842	874	906	938	970	1002	106	j	74	J
523	555	587	619	651	683	715	747	779	811	843	875	907	939	971	1003	107	k	75	K
524	556	588	620	652	684	716	748	780	812	844	876	908	940	972	1004	108	l	76	L
525	557	589	621	653	685	717	749	781	813	845	877	909	941	973	1005	109	m	77	M
526	558	590	622	654	686	718	750	782	814	846	878	910	942	974	1006	110	n	78	N
527	559	591	623	655	687	719	751	783	815	847	879	911	943	975	1007	111	o	79	O
528	560	592	624	656	688	720	752	784	816	848	880	912	944	976	1008	112	p	80	P
529	561	593	625	657	689	721	753	785	817	849	881	913	945	977	1009	113	q	81	Q
530	562	594	626	658	690	722	754	786	818	850	882	914	946	978	1010	114	r	82	R
531	563	595	627	659	691	723	755	787	819	851	883	915	947	979	1011	115	s	83	S
532	564	596	628	660	692	724	756	788	820	852	884	916	948	980	1012	116	t	84	T
533	565	597	629	661	693	725	757	789	821	853	885	917	949	981	1013	117	u	85	U
534	566	598	630	662	694	726	758	790	822	854	886	918	950	982	1014	118	v	86	V
535	567	599	631	663	695	727	759	791	823	855	887	919	951	983	1015	119	w	87	W
536	568	600	632	664	696	728	760	792	824	856	888	920	952	984	1016	120	x	88	X
537	569	601	633	665	697	729	761	793	825	857	889	921	953	985	1017	121	y	89	Y
538	570	602	634	666	698	730	762	794	826	858	890	922	954	986	1018	122	z	90	Z
539	571	603	635	667	699	731	763	795	827	859	891	923	955	987	1019	123	{	91	[
540	572	604	636	668	700	732	764	796	828	860	892	924	956	988	1020	124	:	92	\
541	573	605	637	669	701	733	765	797	829	861	893	925	957	989	1021	125	}	93	]
542	574	606	638	670	702	734	766	798	830	862	894	926	958	990	1022	126	~	94	^
543	575	607	639	671	703	735	767	799	831	863	895	927	959	991	1023	127	■	95	_
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	← DEC.			
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	← ASCII			
High Order X & Y																			

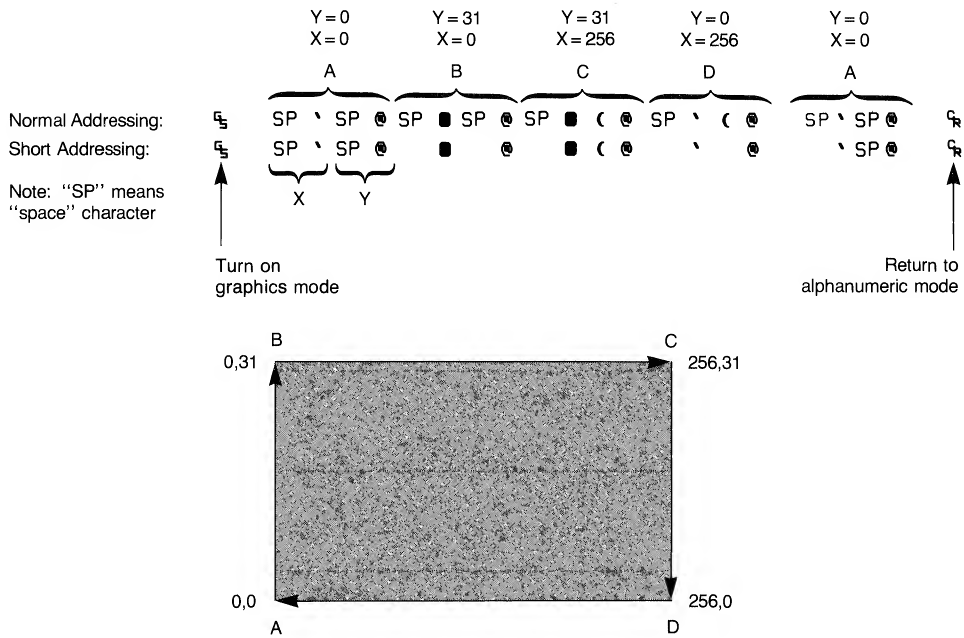
Example: 340Y,70X is found as follows:

340Y = 42 (upper Y) 116 (Lower Y) 70X = 34 (Upper X) 70 (Lower X)

340Y, 70X → * t "F

As an example, to plot the points A (0,0), B (0,31), C (256,31), D (256,0), the sequence would be as shown in figure 3-20.

Figure 3-20. Computer to Terminal Addressing



Text

Text can be placed in either the alphanumeric memory or in the graphics memory.

Text written to alphanumeric memory can be scrolled, edited, erased, etc. without affecting the graphics image. Text written to graphics memory can be scaled, slanted, or rotated (refer to Graphics Text in this section for additional information).

When text is written to the graphics memory, the graphics cursor is moved to indicate where the next character will be stored. (The alphanumeric cursor is only used when data is

stored in the alphanumeric memory.) This differs from terminals that have only one mode for text and display the graphics cursor only when waiting for graphic input from the user.

Scaled and SCL 4014 Modes Text. In Scaled and Scl 4014 modes, the default choice is to write text into graphics memory. When stored in graphics memory, the text size is fixed to allow for 35 lines of text, the text angle is set at 0 degrees and unslanted, and the text origin is set to the left and bottom. These settings allow the Page Full feature to work properly and existing software to run without changes. Even if you don't require the Page Full feature, you can't change the text settings. However, you can redirect the text to the alphanumeric memory by selecting alphanumeric mode, as described previously.

Unscaled and UNS 4014 Modes Text. In Unscaled and Uns 4014 modes, the default text destination is alphanumeric memory. However, it can be sent to graphics memory by entering Graphics mode, as previously explained.

Programming Considerations

When Scaled or Scl 4014 mode is initially selected — whether programmatically, through the Global Configuration menu, at power on, or executing a hard reset — the graphics cursor is turned on and positioned at the upper left-hand corner of the screen display This emulates the 4014 function. Any subsequent selection of either mode will not move the cursor.

Using Color

Color capabilities used in creating a picture are:

1. Selecting palettes.
2. Defining palettes.
3. Loading the selected palette into the system palette.
4. Selecting the drawing mode.
5. Selecting color mixes for the following:

   a. Background pen.
   b. Primary pen.
   c. Secondary pen.
   d. Graphics text pen.
   e. Area boundary pen.

6. Deleting palettes.
7. Setting the selected palette to the power-on colors.
8. Drawing lines.
9. Filling areas.
10. Dithering.
11. Drawing area boundaries.

Defining palettes is normally done as the first step in creating a picture. Selecting a palette, loading the selected palette into the system palette, selecting the background pen, and selecting the drawing mode follow. The remaining operations are used as the need arises during picture creation.

Palettes

A palette consists of eight color mixes, with one of the 64 possible colors assigned to each mix. You can define (select color mixes for) palettes, with up to 15 defined palettes allowed at a given time. User defined palettes are identified by numbers 1 through 15. Palette 0 is the system palette, from which the colors displayed on the screen are taken. You can assign colors to palette 0, thus selecting the colors displayed on the screen.

A palette has three states: selected, defined, and loaded.

Selected — A palette is selected when it is assigned an identification number (1-15).

Defined — A palette is defined when one of the 64 available colors is assigned each of it's eight color mixes.

Loaded — A palette is loaded when the colors assigned it's mixes are assigned the mixes on the system palette.

Before a palette can be defined, it must be the currently selected palette. Only one palette can be selected at a given time, and only the currently selected palette can have its color mixes defined or changed.

To use a color on a defined palette to draw a line, the palette on which the color mix is located must be loaded into the system palette; then the mix (0-7) of the desired color is assigned to the primary pen.

At all times, colors displayed on the screen are the colors of the system palette. If an object in the displayed picture was drawn using mix 5, it will be displayed in the color of mix 5 on the system palette. When a new palette is loaded into the system palette, the colors of the displayed picture will change to the colors of the new palette.

Selecting a Palette. Before the colors of a palette can be changed or the palette loaded into the system palette, the palette must be selected using the following escape sequence:

```
Ec*v <x> p
```

where <x> is the number (0–15) assigned the palette.

Defining a Palette. Defining a palette consists of assigning colors to the mixes on the palette. Before a palette can be defined, it must be selected, as described previously. If no palette is currently selected, it is assumed you mean to define the system palette. Up to 15 palettes, numbered from 1 to 15, can be defined at a given time. Palette 0 (the system palette) can also be defined, but its definition will be lost if another palette is loaded.

If you don't assign colors to all the mixes on the palette being defined, the terminal automatically assigns, to any undefined mixes, the colors assigned to the same-numbered mixes on the system palette.

As an alternative to defining your own palettes, you can use the colors on the system palette. The colors assigned the system palette after a power on or system reset are:

MIX	COLOR
0	Black
1	Red
2	Green
3	Yellow
4	Blue
5	Magenta
6	Cyan
7	White

A palette is defined by assigning a color to each of the eight palette mixes, using the escape sequence:

```
Ec*v <method>m <int>a <int>b <int>c <mix>I
```

where:

`<method>`: 0 = RGB (default)
1 = HSL

`<int>`: Intensity for each of the three color components. The values can range from 0.00 to 1.00, but are reduced to one of four values by the

terminal (refer to "How Color is Generated", Section 2). The default is 0.

a: Red or hue.

b: Green or saturation.

c: Blue or luminosity.

<mix>: The mix (0-7) to which the color is to be assigned.

Example: Set mix 4 to RGB values of 0.5, 0.8, and 0.6:

```
Ec*v 0m .5a .8b .6c 4I
```

Example: Set mix 7 to HSL values of 0.3, 0.1, and 0.5:

```
Ec*v 1m .3a .1b .5c 7I
```

Loading the Selected Palette. Loading the selected palette assigns it's colors to the system palette. Since the colors displayed on the screen are the colors assigned to the system palette, the colors of the displayed picture may change when the selected palette is loaded.

For example, assume the following:

1. Certain lines and areas on the displayed picture were drawn with mix 3.
2. Mix 3 on the system palette is assigned the color red.
3. Mix 3, on the new palette to be loaded into the system palette, is blue.

Then, lines and area fills on the picture which were drawn with mix 3 will change from red to blue when the new palette is loaded into the system palette.

To load the selected palette, use the following escape sequence:

```
Ec*vl
```

As with many other escape sequences, selecting and loading a palette can be combined into one escape sequence, as follows:

```
Ec*v <x> pl
```

where <x> is the number of the palette.

Example: Select and load palette 3.

```
Ec*v 3 pL
```

Selecting the Color for the Background Pen

You can use either the system default color (black—mix 0) for the background pen or select a color of your choice. A color is selected for the background pen by assigning the mix of the desired color on the system palette to the pen.

Once a background color has been selected, it remains so until another mix is assigned to the background pen, the default selections are selected with the "Ec*mr" escape sequence, a hard reset is performed or the power is turned off. After a hard reset or power on, the background pen is assigned color mix 0 (black) on the system palette.

Use the following escape sequence to assign a mix from the system palette to the background pen:

```
Ec*e <x> b
```

where <x> is the mix number, 0-7.

Selecting the Drawing Mode

The drawing mode determines how overlapping lines and filled areas are displayed. Refer to "Drawing Modes for Color Terminals", Section 2, for information on the nature of each drawing mode and how to select it.

Selecting a Color for the Primary Pen

The primary pen is the pen used for drawing and area filling. A color is selected for the primary pen by assigning the mix of the desired color from the system palette to the primary pen. To draw with a color different from the one currently assigned to the primary pen, a new color mix must be assigned the primary pen.

Once a color has been assigned to the primary pen, it remains so until another mix is assigned to the pen, the default selections are selected with the "Ec*mr" escape sequence, a hard reset is performed, or power is turned off. After a hard reset or power on, mix 7 (white) on the system palette is assigned the primary pen. (The default color assigned the background pen is black—mix 0.)

The following escape sequence selects a mix on the system palette for the primary pen:

```
Ec*m <mix>x
```

where <mix> is the mix number.

Selecting a Color for the Secondary Pen

The secondary pen is used in drawing mode Jam2 (mode 4) to draw the points indicated by binary "0's" in the data (binary "1's" are drawn with the primary pen). The default selection (black—mix 0) can be used, or you can select a mix of the desired color from the system palette using the following escape sequence:

```
Ec*m <mix> y
```

where <mix> is the mix number.

Selecting a Color for the Text Pen

Graphics text is created using the graphics text pen. The text pen is assigned a color mix from the system palette, using the following escape sequence:

```
Ec*n <mix> x
```

where <mix> is the mix number.

Selecting/Disabling the Area Boundary Pen

The boundary pen is used to outline the boundary of a filled area. It can be either active or disabled; the default condition is disabled. To be active, it must be assigned a color mix from the system palette using the following escape sequence:

```
Ec*m <mix>h
```

where <mix> is the mix number.

To disable the pen, use the following escape sequence:

```
Ec*m h
```

To be used, the pen must be lowered; to be temporarily disabled, it is lifted. Refer to "Drawing Area Boundaries", later in this section for the escape sequences for lowering and lifting the pen.

Deleting Palettes

You can delete any defined palette or delete all defined palettes, as follows:

`Ec*v <x> d`	Delete palette <x>.
`Ec*ve`	Delete all defined palettes.

Setting the Selected Palette to the Power On Colors

The selected palette can be set to the colors specified for power on or hard reset conditions with the following escape sequence:

```
Ec*vr
```

Drawing Lines

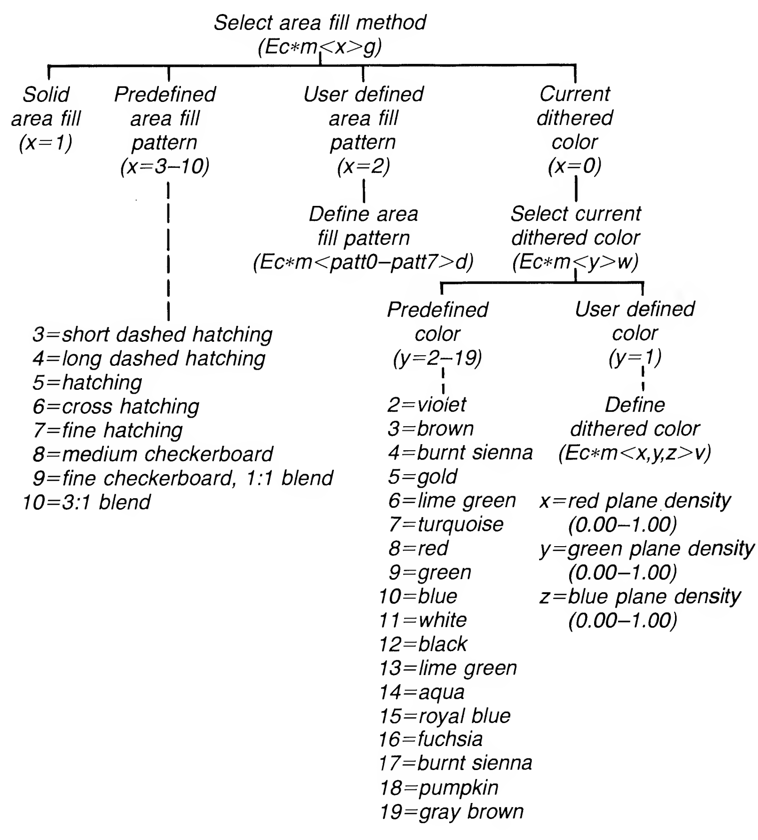
For color terminals, lines are drawn according to the same rules as for monochrome terminals, except that the drawing modes are entirely different. Refer to the discussion on color in Section 2 for information on the color drawing modes.

Filling Areas

Filling areas for color terminals is done according to the same rules as for monochrome terminals except that, in addition to solid, predefined, and user defined area fill methods, a fourth method — dithering — is added.

If a dithered color is to be used, either the currently selected dithered color must be used, one must be selected from 18 predefined colors, or you must define a color. Figure 3-21 shows the selections to be made for area filling, together with the associated escape sequences.

Figure 3-21. Area Fill Selection Chart



When an area is specified for filling, the terminal uses the area fill method currently selected. This is true also for the predefined area fill pattern, the user defined area fill pattern, and the currently selected dithered color.

Solid area filling uses the primary pen. Area fill patterns, both predefined and user defined, are made using the colors from the system palette.

Dithering

Dithering is a means of achieving many colors for area filling at the expense of some resolution. This reduced resolution may require that, for clarity, single-pixel-width lines and graphics text superimposed on a dithered background be given extra width or size.

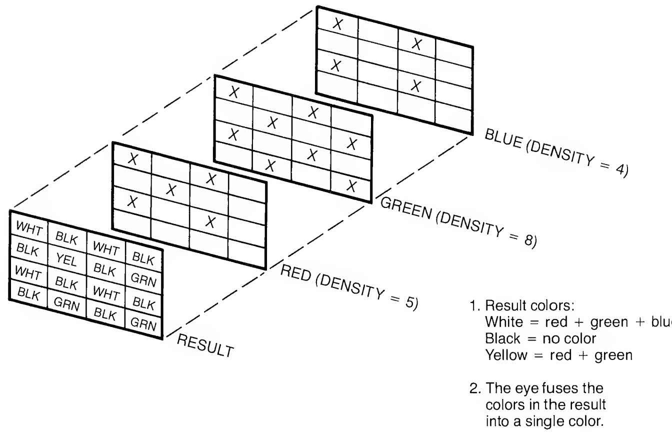
Dithering, which is used only for area filling, depends on the ability of the eye to integrate closely packed pixels of different colors into a single color.

For dithering, the terminal firmware divides each of the red, green, and blue color planes into arrays of 4X4 dots, so that there are 16 dots in an array. By assigning a "density" to each color plane, the user can select the number of turned-on dots in each array in the plane. The density, along with a fixed priority sequence for turning on dots which is imposed by the firmware, determines which dots will be turned on in the arrays. The turn-on priority sequence is as shown below:

1	13	4	16
9	5	12	8
3	15	2	14
11	7	10	6

Figure 3-22 illustrates dithering.

Figure 3-22. How a Color is Produced by Dithering



You can use either a predefined dithered colors or define your own, but to make it available for area filling, escape codes must be used to select dithering as the area fill method and the color as the current dithered color.

Predefined Dithered Colors. The terminal has 18 predefined dithered colors which may be used to fill an area. The predefined dithered colors closely match the colors of the pens used on Hewlett Packard's plotters.

The following escape sequence is used to select a predefined dithered color as the current dithered color:

```
Ec*m <x>w
```

where <x> is a number which selects the color as shown below:

<x>	COLOR	DENSITIES R	G	B
2	Violet	0.50	0.13	0.06
3	Brown	0.50	0.13	0.00
4	Burnt Orange	1.00	0.25	0.00
5	Gold	1.00	0.75	0.00
6	Lime	0.50	0.50	0.13
7	Turquoise	0.00	0.50	0.50
8	Red	0.50	0.00	0.00
9	Green	0.00	0.50	0.13
10	Blue	0.00	0.00	0.75
11	White	1.00	1.00	1.00
12	Black	0.00	0.00	0.00
13	Lime Green	0.25	0.50	0.13
14	Aqua	0.00	0.50	0.50
15	Royal Blue	0.00	0.00	0.50
16	Fuchsia	0.50	0.00	0.50
17	Burnt Sienna	0.50	0.13	0.06
18	Pumpkin	1.00	0.50	0.00
19	Gray Brown	0.50	0.25	0.25

User Defined Dithered Color. The current dithered color may be defined by the user. To do this, the user uses the following escape sequence to select densities for each of the three color planes.

```
Ec*m <x>,<y>,<z>v
```

where <x>, <y>, and <z> are decimal fractions in the range 0.00 to 1.00, inclusive, which select the densities for the red, green, and blue color planes, respectively. The terminal converts the decimal number to its nearest equivalent in 16ths to determine the number of dots to use.

Example: Prepare to fill an area with dithered orange:

1. Define the current dithered color by selecting the densities for the three color planes:

   `Ec*m1,.5,0V`

2. Select the user defined dithered color as the dithered color:

   `Ec*m1W`

3. Select the dithered color as the area fill method:

   `Ec*m0G`

Drawing Area Boundaries

If desired, the boundary of a polygon can be outlined in a color different from the area fill color. To do this, a boundary pen color must be selected. In addition, it is also necessary to ensure that the pen is lowered to draw a boundary line.

Selecting the Boundary Pen. The boundary pen can be selected and disabled. Once selected, it remains so until reselected with another select command, or disabled. Selecting the boundary pen consists of assigning it a color mix from the system palette. The escape sequence is as follows:

```
Ec*m <mix>h
```

where `<mix>` is the number of the mix, on the system palette, with the desired color. It can be an integer in the range −32768 through 32767; the three least significant bits are used to select the mix (0–7).

Lift/Lower Boundary Pen. The boundary pen can be lifted and lowered. For use, it must be lowered. The escape sequence is as follows:

```
Ec*p<x>
```

where:

<x>	ACTION
u	Lift pen.
v	Lower pen.

4 Alphanumeric Display Control

Introduction

The alphanumeric display portion of the terminal consists of alphanumeric display memory and the display screen. Alphanumeric display memory is the part of terminal memory assigned to alphanumeric data, for display on the screen. It can hold several "pages" (screenfuls) of data.

The display screen consists of 27 lines; each line contains space for 80 characters. The first 24 lines are used to display one page of alphanumeric display memory, rows 25 and 26 display the function key labels, and row 27 contains information on the terminals operational status.

In alphanumeric (non-graphic) mode, the alphanumeric cursor is always displayed on the screen as either a blinking underline or a blinking box, depending on the selection in the "`Cursor Type`" field of the Global Configuration menu. The cursor indicates the line and character cell currently selected for any display-changing action, such as inserting or deleting a character or line.

Note Throughout this section, "display memory" refers to alphanumeric display memory, not graphics memory.

You can perform the following display control operations either locally from the keyboard or remotely from a program executing in a host computer.

- Move the cursor up, down, left, or right.
- Roll the displayed data up or down on the screen. When a roll-up or roll-down operation forces data off the top or

bottom edge of the screen, additional data rolls onto the screen at the opposite edge.

- Roll displayed data left or right. When this is done, data forced off one side of the screen is replaced by data which rolls onto the screen at the opposite side. This operation can be performed only when a line length greater than 80 characters is selected in the "`Columns`" field of the Global Configuration menu.
- Change the data displayed on the screen to the next, previous, next left, or next right "page" of data in display memory. A page is a sequence of 24 lines of data (the number of lines that can be displayed on the screen).
- Set or clear a left and right margin.
- Set or clear one or more tab stop positions.
- Move the cursor forward to the next tab stop position or backward to the preceding tab stop position.
- Enable or disable the inverse video, half bright, underline, blinking, and/or security display enhancements.
- Change from one character set to another (base, Math, Line Drawing, Bold, or Italics).
- Create data entry forms containing protected, unprotected, and transmit-only fields.

In addition, you can do the following screen edit operations either locally or remotely:

- Delete or clear all characters from the current cursor position through the end of display memory.
- Delete the line containing the cursor (subsequent lines are rolled up).
- Change the characters in the line containing the cursor, from the cursor to the end of the line, to blanks.
- Delete the character at the current cursor position (this can be done with or without character wraparound from the next line).

- Insert a blank line immediately preceding (above) the line currently containing the cursor.
- Enable or disable Insert Character mode. When this editing mode is enabled, succeeding characters entered through the keyboard or received from the host computer are inserted to the left of the character at the current cursor position. This editing mode can be enabled either with or without character wraparound to the next line.

Cursor Controls

The following topics describe how to alter the cursor/data relationship either manually (by using the cursor control keys) or programmatically (by using escape sequences).

Turning Alphanumeric Cursor On and Off

The alphanumeric cursor can be turned on and off using the following escape sequences:

ON: `Ec*dQ`

OFF: `Ec*dR`

Home Up

Pressing the ⇱ key moves the cursor to the left margin in the top row of the screen and rolls the text down as far as possible so that the first line in display memory appears in the top row of the screen.

When Format mode is enabled, the ⇱ key also rolls the text down as far as possible but leaves the cursor positioned at the beginning of the first unprotected field. If no fields have been defined, the cursor will appear at the first column of the first row on the screen.

To perform this function programmatically, use either of the following escape sequences:

```
Ec h
```
or
```
Ec H
```

When Memory Lock is enabled, the ⍄ key rolls all unlocked text down as far as possible below the block of locked text. This leaves the cursor positioned at the left margin of the first unlocked row instead of the first row in display memory.

When both Format and Memory Lock modes are active simultaneously, the cursor will move to the first unprotected field on the screen (including the locked area), after rolling all the text down.

Note If Memory Lock mode is on and the cursor is within the locked area, the ⍄ key, after rolling the text down, will cause the cursor to go to the left margin of the first line of text under the locked area.

When Format mode is enabled, you may "home up" the cursor programmatically. However, the cursor will be positioned at the beginning of the first unprotected field or transmit-only field, whichever occurs first. This is done by using either of the following escape sequences:

```
Ec h
```
or
```
Ec H
```

Home Down

Pressing the Shift and ⍄ keys, simultaneously, moves the cursor to the left margin in the bottom line of the screen and rolls the text in alphanumeric display memory up as far as possible so that the last "typed on" or "used" line in alphanumeric display memory appears immediately above the cursor position.

In a configuration menu, the cursor will go to the left margin in the line just after the last used line in memory.

To perform this function programmatically, use the following escape sequence:

```
Ec F
```

Move Cursor Up

Each time you press the ▲ key, the cursor moves upward one row in the current column. If you hold the key down, the cursor movement continues row-by-row until the key is released. When the cursor is in the top screen row, pressing this key moves the cursor to the bottom screen row, same column.

To perform this function programmatically, use the following escape sequence:

```
Ec A
```

Move Cursor Down

Each time you press the ▼ key, the cursor moves down one row in the current column. If you hold the key down, the cursor movement continues row-by-row until the key is released. When the cursor is in the bottom screen row, pressing this key moves the cursor to the top screen row, same column.

To perform this function programmatically, use the following escape sequence:

```
Ec B
```

Move Cursor Right

Each time you press the ▶ key, the cursor moves one column to the right. If you hold the key down, the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the rightmost column of the screen, pressing this key moves the cursor to the leftmost

screen column in the next lower row (from the rightmost column in the bottom row of the screen, the cursor moves to the leftmost column in the top row of the screen).

To perform this function programmatically, use the following escape sequence:

```
Ec C
```

Move Cursor Left

Each time you press the ◄ key, the cursor moves one column to the left. If you hold the key down, the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the leftmost column on the screen, pressing this key moves the cursor to the rightmost column in the next higher row (from the leftmost column in the top row, the cursor moves to the rightmost column in the bottom row on the screen).

To perform this function programmatically, use the following escape sequence:

```
Ec D
```

Screen Addressing

To move the cursor to any character position currently visible on the screen, use any of the following escape sequences:

```
Ec&a <column number> x <row number> Y

Ec&a <row number>    y <column number> X

Ec&a <column number> X

Ec&a <row number>    Y
```

where:

`<column number>` —is a decimal number specifying the screen column to which you wish to move the cursor. Zero specifies the leftmost column.

`<row number>` —is a decimal number specifying the screen row to which you wish to move the cursor. Zero specifies the top row.

When using the above escape sequences, the data visible on the screen always remains unchanged.

If you specify only a `<column number>`, the cursor remains in the current row. Similarly, if you specify only a `<row number>`, the cursor remains in the current column.

If you specify a column number greater than the right boundary of the screen, the cursor will move to the right boundary. If you specify a row number greater than the last displayed row, the cursor will move to the last displayed row.

Example: The window contains 24 rows. The following escape sequence moves the cursor to the 20th column of the 7th row on the screen:

```
Ec&a6y19X
```

Memory Addressing

You can specify any location in display memory by supplying memory-relative row and column coordinates. (Row selection is disabled when Memory Lock mode is on, but selection can still be done.) To move the cursor to another character position using memory addressing, use any of the following escape sequences:

```
Ec&a <column number> c <row number> R

Ec&a <row number>    r <column number> C
```

```
Ec&a <column number> C

Ec&a <row number>    R
```

where:

`<column number>` —is a decimal number specifying the column (in display memory) of the location at which you want the cursor positioned. Zero specifies the first (leftmost) column in display memory.

`<row number>` —is a decimal number specifying the row (in display memory) of the location at which you want the cursor positioned. Zero specifies the first (top) row in display memory.

When using the above escape sequences, the data visible in the window will (if necessary) be rolled up or down or moved left or right to position the cursor at the specified location. The cursor and data movement will occur as follows:

- If the specified location lies within the boundaries of the window, the cursor moves to that position and the data on the screen remains unchanged.
- If the specified column location exceeds the right boundary of the window and a line length greater than 80 columns is selected in the "`Columns`" field of the Global Configuration menu, text is moved left until the specified column occupies the rightmost screen column. Also, the cursor moves to the rightmost column in the window.
- If the absolute row location is less than that of the top line currently visible on the screen, the cursor moves to the specified column in the top row of the window and the text rolls downward until the specified row appears in the top line of the window.

- If the absolute row location exceeds that of the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the window and the text rolls upward until the specified row appears in the bottom line of the window.

If you specify only a `<column number>`, the cursor remains in the current row. Similarly, if you specify only a `<row number>`, the cursor remains in the current column.

Note

If you specify a row greater than the last memory row, lines will be discarded from the top of display memory to create as many lines as necessary, at the bottom of display memory, to locate the cursor in the line you have specified.

Example: Display memory contains 54 rows. The following escape sequence moves the cursor (and rolls the text if neccessary) so that it is positioned at the character in the 95th column of the 30th row in display memory:

```
Ec&a29r94C
```

Cursor-Relative Addressing

You can specify the location of any character within display memory by specifying the row and column relative to the current cursor position (this function is disabled for rows when Memory Lock mode is on).

To move the cursor to another character position using cursor-relative addressing, use any of the following escape sequences:

```
Ec&a  +/- <column number> c  +/- <row number> R

Ec&a  +/- <column number> x  +/- <row number> Y

Ec&a  +/- <row number> r  +/- <column number> C

Ec&a  +/- <row number> y  +/- <column number> X
```

```
Ec&a  +/- <column number> C

Ec&a  +/- <row number> R

Ec&a  +/- <column number> X

Ec&a  +/- <row number> Y
```

where:

`<column number>`	is a decimal number specifying the relative column to which you wish to move the cursor. A positive number specifies how many columns to the right you wish to move the cursor; a negative number specifies how many columns to the left.
`<row number>`	is a decimal number specifying the relative row to which you wish to move the cursor. A positive number specifies how many rows downward you wish to move the cursor; a negative number specifies how many rows upward.

When using the "R" parameter for row selection, the data will (if necessary) be rolled up or down to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If the specified location lies within the boundaries of the window, the cursor moves to that position and the data on the screen remains unchanged.
- If the specified row is less than the first row of the window (first displayed row), the cursor moves to the topmost row in the window. (Text rolls down until the specified row is the first row in the window.)

- If the specified row is greater than the last row in the window, the cursor moves to the bottommost row in the window. (Text rolls up until the specified row is the bottommost row in the window.)
- If, when using the "R" parameter for row selection, the specified cursor-relative row precedes the top line of the window currently visible on the screen, the cursor moves to the specified column in the top row of the window and the text rolls downward until the specified row appears in the top line of the window.
- If, when using the "R" parameter for row selection, the specified cursor-relative row follows the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the window and the text rolls upward until the specified row appears in the bottom line of the window.
- If, when using the "Y" parameter for row selection, the selected row is greater than the last displayed row, the cursor will move to the last displayed row.

When using the "C" parameter for column selection, the data will (if necessary) be moved left or right to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If the specified location lies within the boundaries of the window, the cursor moves to that position and the data on the screen remains unchanged.
- If the specified column is less than the left boundary of the window, the cursor moves to the leftmost column in the window. (The data shifts right until the specified column is the leftmost column in the window.)
- If the specified column exceeds the right boundary of the window, the cursor moves to the rightmost column in the window. (The data shifts left until the specified column is the rightmost column in the window.)
- If, when using the "C" parameter for column selection, the specified cursor-relative column is less than the leftmost column of the window, the cursor moves to the

specified row in the leftmost column of the window and the text moves right until the specified column appears in the leftmost column of the window.

- If, when using the "C" parameter for column selection, the specified cursor-relative column is greater than the rightmost column of the window, the cursor moves to the specified row in the rightmost column of the window and the text moves left until the specified column appears in the rightmost column of the window.
- If, when using the "X" parameter for column selection, the selected column is greater than the rightmost displayed column, the cursor will move to the rightmost displayed column.

If you specify only a column number, the cursor remains in the current row. Similarly, if you specify only a row number, the cursor remains in the current column.

Example: The following escape sequence moves the cursor (and rolls the text if necessary) so that it is positioned at the character residing 15 columns to the right and 25 rows above the current cursor position in display memory:

```
Ec&a+15c-25R
```

Combining Addressing Methods

You may use a combination of screen, memory, and cursor-relative coordinates within a single escape sequence.

Example: Move the cursor (and roll the text if necessary) so that it is positioned at the character in the 70th column of the 18th row below the current cursor position.

```
Ec&a69c+18R
```

Example: Move the cursor so that it is positioned at the character 15 columns to the left of the current cursor position in the 4th row currently visible on the screen.

```
Ec&a-15c3Y
```

Example: Move the cursor (and roll the text up or down, if necessary) so that it is positioned at the character in the 10th column of row 65 in display memory.

```
Ec&a9c64R
```

Cursor Position Sensing

The current cursor position can be sensed by a program in either memory-relative or screen-relative coordinates. The procedure is for the program to send the appropriate escape sequence, followed by a request for input from the terminal (INPUT command, in BASIC). The terminal responds with the cursor position.

Memory-Relative Cursor Sensing. The following example illustrates sensing the cursor position, in memory-relative coordinates, when the cursor is at column 20, row 40 in display memory.

computer: `Ec a`

terminal: `Ec&a020c040R`

Screen-Relative Cursor Sensing. The following example illustrates sensing the cursor position, in screen-relative coordinates. The cursor is at column 20, row 40 in display memory, but screen row 0 begins at display memory row 35.

computer: `` Ec ` ``

terminal: `Ec&a020x005Y`

Window Control

The window can be positioned in display memory to display selected segments of display memory. Movement can be vertical or horizontal in row, column, or page increments. The window can be moved horizontally only if a line length greater than 80 columns is selected in the "Columns" field of the Global Configuration menu.

The horizontal size of a page is 80 characters; it's vertical size is 24 lines. In Memory Lock mode, it's vertical size is 24 lines minus the number of locked rows.

Turning On and Off the Alphanumeric Window and Display

Escape sequences are available for turning on and off the alphanumeric window (the top 24 lines of the display) and the alphanumeric display (all 27 lines of the display). Turning off the window leaves the function keys and the status line displayed; turning off the display blanks the entire screen.

To turn the window on and off, use the following escape sequences:

ON: `Ec&w12E`

OFF: `Ec&w13F`

To turn the display on and off, use the following escape sequences:

ON: `Ec*de`

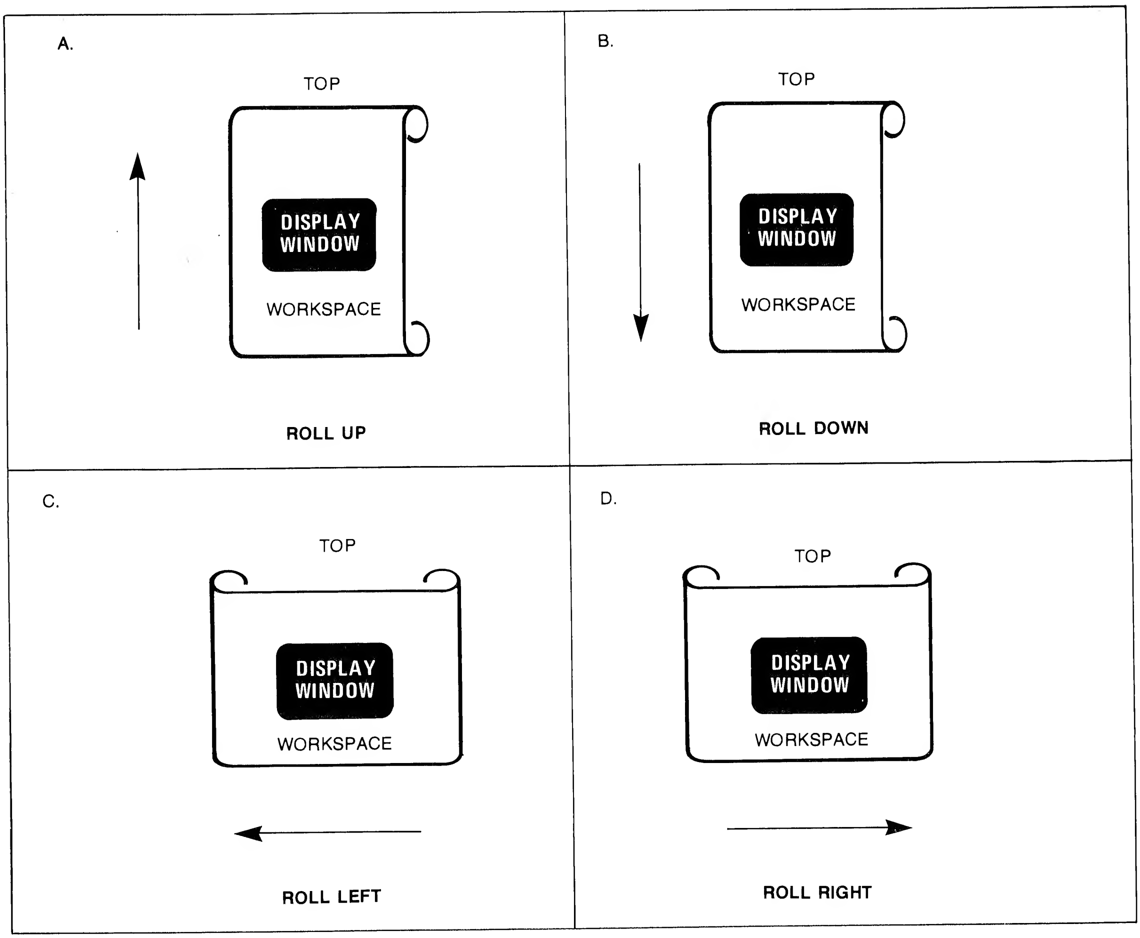
OFF: `Ec*df`

Roll Text Up

Each time you press the [Shift] and [▲] keys together, the text rolls up one row. The top row rolls off the screen and the remaining data rolls up one line on the screen, with a new line of data rolling from display memory to the bottom line of the window. If you hold this key down, the text continues to roll upward until you release the key or until the final line of data in display memory appears in the top row

of the window. In the latter case, pressing or holding down the key has no further effect. The "roll up" function is illustrated in figure 4-1.

Figure 4-1. The "Roll" Data Functions



When a configuration menus or User Key menu is displayed, this key is disabled. In Memory Lock mode, the unlocked text rolls around the locked text, as if the bottom line of the locked text is the top of the window.

To programmatically roll the text up one row, use the following escape sequence:

```
Ec S
```

To roll the text up a selected number of lines, the following sequence can be used:

```
Ec&r<rows>U
```

where:

`<rows>`	is the number of rows to be rolled up.

Roll Text Down

Each time you press the [Shift] and [▼] keys together, the text rolls down one row on the screen. The bottom row rolls off the screen and the remaining data rolls down one line, with a new line of data rolling from display memory to the top line of the window. If you hold this key down, the text continues to roll down until you release the key or until the first line of data in display memory appears in the top row of the screen. In the latter case, pressing or holding down the key has no further effect. The "roll down" function is illustrated in figure 4-1.

This key is disabled when a configuration menu or the User Key menu is displayed. If the [Shift] and [▼] keys are pressed simultaneously in Memory Lock mode, any text preceding the locked text will be scrolled down, a line at a time, and inserted between the locked text and the first line following the locked text.

To programmatically roll the text down one row, use the following escape sequence:

```
Ec T
```

To roll the text down a selected number of rows, the folowing sequence can be used:

```
Ec&r<rows>D
```

where:

<rows> is the number of rows the text is to be rolled down.

Roll Text Right

This operation can be performed only if a line length greater than 80 is selected in the "Columns" field of the Global Configuration menu. Each time you press the [Shift] and [▶] keys together, the text moves right one column. The right column moves off the screen, and a new column moves onto the screen from the left. If you hold the key down, the text continues to move right, column-by-column, until you release the key or the leftmost column of display memory appears in the leftmost screen column. In the latter case, holding down the key has no further effect.

To move the text right a selected number of columns, the following escape sequence can be used:

```
Ec&r<col>R
```

where:

<col> is the number of columns the text is to be moved.

Roll Text Left

This operation can be performed only when a line length greater than 80 is selected in the "Columns" field of the Global Configuration menu. Each time you press the [Shift] and [◀] keys together, the text moves left one column. The left column moves off the screen, and a new column moves onto the screen from the right. If you hold the key down, the text continues to move right, column-by-column, until you release the key or the rightmost column in display memory appears in the rightmost screen column. In the latter case, holding down the key has no further effect.

To move the text left a selected number of columns, the following escape sequence can be used:

```
Ec&r<col>L
```

where:

<col>	is the number of columns the text is to be moved.

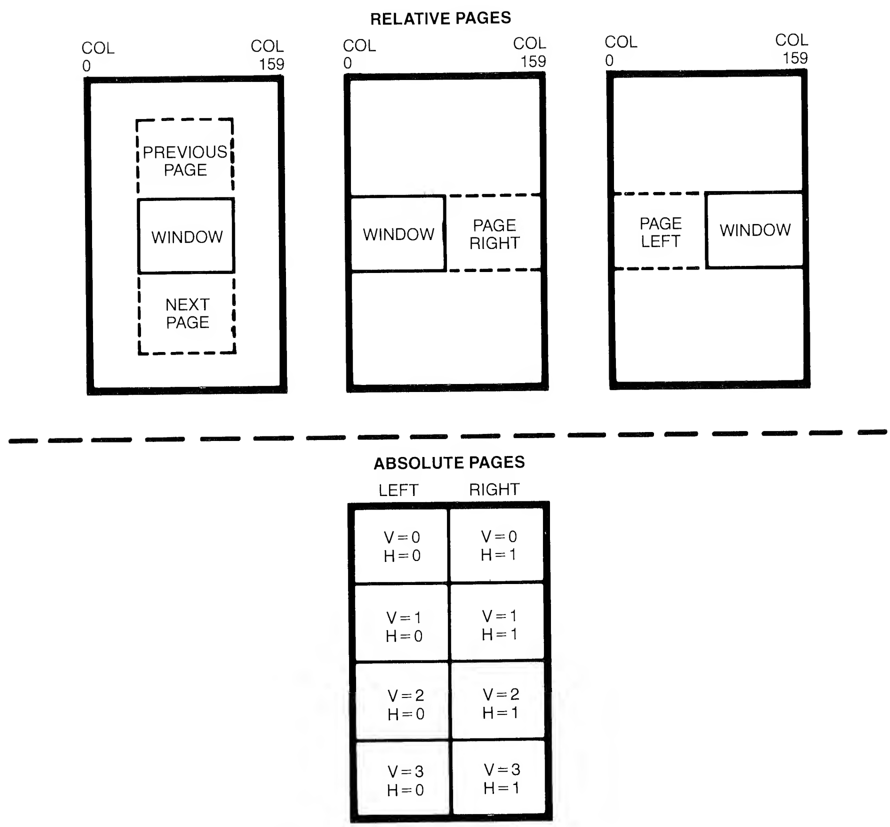
Displaying Selected Pages

Data in alphanumeric memory is displayed on the screen in blocks known as "pages". Except in Memory Lock mode, a page is one screenful (24 lines by 80 columns), which is also the size of the window. In Memory Lock mode, the page size is equal to the number of "unlocked" lines in the window.

At the completion of any page function, the cursor is positioned at the left margin in the top line on the screen.

The concept of display "pages" is illustrated in figure 4-2.

Figure 4-2. Page Concepts



Next Page. Pressing the [Next] key displays the next page of data, replacing the current page on the screen. If you hold the key down, the operation is repeated until you release the key or until the final line of text in display memory appears as the top line on the screen. In the latter case, pressing or holding down the key has no further effect.

When a configuration menu or the User Key menu is displayed, this key is disabled.

To perform the "next page" function programmatically, use the following escape sequence:

```
Ec U
```

To do a selected number of "next page" operations, the following escape sequence can be used:

```
Ec&r +<pages>V
```

where:

<pages>	is the number of "next page" operations to be performed.

Previous Page. Pressing the [Prev] key rolls the text in display memory down so that the previous page of data replaces the current page in the window. If you hold the key down, the operation is repeated until you release the key or until the first line in display memory appears in the top line of the screen. In the latter case, pressing or continuing to hold down the key has no further effect.

When a configuration menu or the User Key menu is displayed, this key is disabled.

To perform the "previous page" function programmatically, use the following escape sequence:

```
Ec V
```

To do a selected number of "previous page" operations, the following escape sequence can be used:

```
Ec&r -<pages>V
```

where:

`<pages>` is the number of "previous page" operations to be performed.

Page Left. The page to the left of the current page can be displayed using the following escape sequence:

```
Ec&r -<pages>H
```

where:

`<pages>` is the number of "page left" operations to be performed.

Effectively, only the number "1" can be used for `<pages>` in the sequence, because display memory has an effective width of 160 characters (two pages wide). If there is less than one page width (80 characters) between the leftmost column of the current window and the left edge of display memory, the leftmost column in display memory will become the leftmost column displayed on the screen.

The text can be moved maximum right, so that the leftmost column in display memory becomes the left column on the screen, by pressing the [Shift] and [Prev] keys together.

Page Right. The page to the right of the current page can be displayed using the following escape sequence:

```
Ec&r +<pages>H
```

where:

`<pages>` is the number of "page right" operations to be performed.

Effectively, only the number "1" can be used for `<pages>` in the sequence, because display memory has an effective width of 160 characters (two pages wide). If there is less than one page width (80 characters) between the rightmost

column of the current window and the right edge of display memory, the rightmost column in display memory will become the rightmost column displayed on the screen.

The text can be moved maximum left, so that the rightmost column in display memory becomes the right column on the screen, by pressing the Shift and Next keys together.

Displaying a Selected Absolute Page. A selected page, as addressed absolutely (figure 4-2), can be displayed using the following escape sequences:

```
Ec&r<page>V

Ec&r<page>H
```

where:

`<page>V`	is the vertical sequence number of the page.
`<page>H`	is the horizontal sequence number of the page.

Setting and Clearing Margins

You can redefine the left and/or right margin. These margins affect the cursor positioning for certain functions (such as carriage return, home up, home down, etc.) and establish operational bounds for the insert character and delete character functions. In addition, the left margin is always an implicit tab stop. Data to the left of the left margin or to the right of the right margin is still accessible. Data transfers from display memory to a host computer or to a printer are performed without regard to margins. Format mode, when enabled, also operates without regard to margins.

When you are entering data through the keyboard and the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (this characteristic can be disabled using the "InhEolWrp" field on the Terminal Configuration menu; refer to Section 6). When you press [Return], the cursor moves to the left margin in the current line if Auto Line Feed mode is disabled or to the left margin in the next lower line if Auto Line Feed mode is enabled.

When data is being received from a host computer, it enters display memory only within the defined margins. When the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (as mentioned above, this operating characteristic can be disabled through the use of the "InhEolWrp" configuration parameter). When an ASCII <CR> control code is received, the cursor always moves to the left margin in the current line regardless of whether or not Auto Line Feed mode is enabled.

From the Keyboard

From the keyboard, you set and clear the margins using the "margins/tab/col" set of function keys. To get to that set, use the following keystroke sequence:

[System], **margins/tabs/col**

To change the left or right margin, move the cursor to the desired column and press the appropriate function key (**LEFT MARGIN** or **RIGHT MARGIN**). To reset the left margin to column 1 and the right margin to the line length selected in the "Columns" field of the Global Configuration menu, press **CLR ALL MARGINS**.

If you attempt to set either margin incorrectly with relation to the other (e.g., the right margin to the left of the left margin), the terminal rejects it with an audible "beep".

From a Program

From a program executing in a host computer, you set and clear the margins using the following escape sequences:

SET LEFT MARGIN:	`Ec 4`
SET RIGHT MARGIN:	`Ec 5`
CLEAR ALL MARGINS:	`Ec 9`

The first two escape sequences set the left and right margin (respectively) at the current cursor position. Therefore, before using them, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

Setting and Clearing Tabs

Within display memory you can define a series of tab stops to which you can move the cursor using the tab and back tab functions (described as separate topics later in this section).

From the keyboard, you set and clear tab stops using the "margins/tabs/col" set of function keys. To get to that set, use the following keystroke sequence:

System, margins/tabs/col

To set a tab stop, move the cursor to the desired column and then press the SET TAB key. To clear a tab stop, move the cursor to the tab stop and then press CLEAR TAB. To clear all existing tab stops, press CLR ALL TABS. Note that the left margin is always an implicit tab stop and is not affected by the CLR ALL TABS key.

TAB stops that do not lie within the area bounded by the left and right margins are ignored when the tab or back tab functions are performed.

From a program executing in a host computer, you set and clear tab stops using the following escape sequences:

SET TAB:	`Ec 1`
CLEAR TAB:	`Ec 2`
CLEAR ALL TABS:	`Ec 3`

The first two escape sequences set and clear (respectively) a tab stop at the current cursor position. Therefore, before using them, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

Tab

From the keyboard, you can move the cursor ahead to the next subsequent tab stop using the [Tab] key. In Format mode, pressing the [Tab] key once moves the cursor to the beginning of the next unprotected field (transmit only fields are ignored). Tabbing from the last field moves the cursor to the beginning of the first field. Tab acts similarly in the User Key menu and configuration menus.

If Memory Lock is on, Format mode is active, the cursor is within the locked area, and the next (and only other) unprotected field has been rolled behind the locked rows, a tab will first roll the hidden field into view. The next tab will move the cursor to the first column of the newly revealed field.

From a program executing in a host computer, you can move the cursor ahead to the next tab stop issuing either an ASCII <HT> control code (decimal 9; Control "I") or the following escape sequence:

```
Ec I
```

Tab commands received from the host when the terminal is in Format mode positions the cursor to the next unprotected or transmit-only field, whichever is encountered first.

Tab stops that do not lie within the area bounded by the left and right margins are ignored by the tab function.

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the right of the rightmost tab stop, the tab function moves the cursor to the left margin in the next lower line. When the cursor is positioned to the left of the left margin, however, the tab function advances the cursor to the first explicit tab stop (not the left margin) in the line (or to the left margin in the next lower line if no explicit tab stops are defined).

Back Tab

From the keyboard you can move the cursor backward to the previous tab stop using the [Shift] and [Tab] keys.

In Format mode, the cursor, if within a field, will move to the beginning of the field; otherwise it will move to the first character of the previous unprotected field (transmit-only fields are ignored). The back tab feature is enabled in the User Key and configuration menus.

From a program executing in a host computer you can move the cursor backward to the previous tab stop using the following escape sequence:

```
Ec i
```

Back tab commands received from the host when the terminal is in Format mode position the cursor to the previous unprotected or transmit-only field, whichever is encountered first, if the cursor is at the beginning of an unprotected or transmit-only field or in a protected field.

When not in Format mode, tab stops that do not lie within the area bounded by the left and right margins are ignored by the back tab function.

The left margin is treated as a tab stop. When the cursor is positioned at or to the left of the left margin, the back tab function moves the cursor to the rightmost tab stop in the next higher line.

Performing a back tab with the cursor on the left margin of the first row on the screen (or the first unlocked row if Memory Lock mode active) will cause the text to roll down one line, if there is a row above the first row.

Edit Operations

You can edit data displayed on the screen by simply over-striking the old data. In addition, the terminal provides the following edit functions which can be enabled and disabled either manually, by using the edit control keys, or programmatically, by using escape sequences:

- Insert Line.
- Delete Line.
- Insert Character.
- Delete Character.
- Clear Display.
- Clear Line.

Insert Line

When you use the insert line edit function, the text line containing the cursor and all text lines below it roll downward one line, a blank line is inserted in the screen row containing the cursor, and the cursor moves to the left margin of the blank line. Note that when Memory Lock mode is active, inserting a line within the locked area of the screen does not change the size of the locked area.

From the keyboard, each time you press the [Insert line] key the terminal inserts one blank line. If you hold the key down, the terminal continues to insert blank lines until the key is released.

This function is disabled in Format mode, and is disabled in the configuration menus and User Key menu.

From a program executing in a host computer, you insert a blank line at the current cursor position using the following escape sequence:

```
Ec L
```

Delete Line

When you use the delete line edit function, the text line containing the cursor is deleted from display memory, all text lines below it roll upward one row, and the cursor moves to the left margin. Note that when Memory Lock mode is active, deleting a line within the locked area does not change the size of the locked area.

From the keyboard, each time you press the [Delete line] key the terminal deletes one line of text. If you hold the key down, the terminal continues to delete text lines until the key is released or until there are no subsequent text lines remaining in display memory. In the latter case, pressing or continuing to hold down this key has no further effect.

This function is disabled in Format mode, and is disabled in the configuration and User Key menus.

From a program executing in a host computer, you delete the text line at the current cursor position using the following escape sequence:

```
Ec M
```

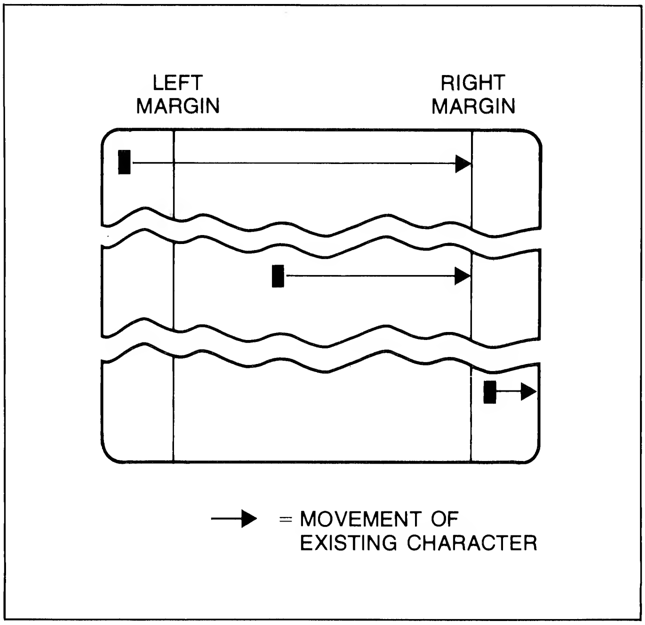
Insert Character

When the insert character editing function is enabled, characters entered through the keyboard or received from the host computer are inserted into display memory at the cursor position. Each time a character is inserted, the cursor and all characters from the current cursor position through the right margin move one column to the right. Characters that are forced over the right margin are lost. When the cursor reaches the right margin, it moves to the left margin in the next lower line and the insert character function continues from that point.

This edit function is meant to be used within that portion of display memory bounded by the left and right margins. If you position the cursor to the left of the left margin, the insert character function works as described above. If you position the cursor beyond the right margin, however, the insert character function affects those characters between the current cursor position and the right boundary of the workspace (as selected in the "Columns" field of the Global Configuration menu). In such a case, when the cursor reaches the right boundary of the workspace, it moves to the left margin in the next lower line and the insert character function continues from that point as described in the first paragraph above.

The movement of existing characters during an "insert character" editing operation is illustrated in figure 4-3.

Figure 4-3. Character Insert with Margins



When Format mode is off, any unprotected, transmit-only, alternate character set, and/or video enhancement fields to the right of the cursor move to the right with the displayable characters. If the cursor is positioned within any such field, the insert character function extends the range of the field by one position for each character inserted. Block terminators at or to the right of the cursor position move to the right along with the displayable characters. Non-displaying terminators to the right of the cursor also move to the right along with the displayable characters; a non-displaying terminator at the cursor position, however, remains at that position.

When Format mode is on and the cursor is positioned within an unprotected or transmit-only field, the insert character function affects only those characters from the cursor position through the end of the current field. Block terminators and non-displaying terminators are treated the same as when Format mode is off. If the cursor is not within an unprotected field, it automatically moves to the first character position of the next subsequent unprotected field when the first character is inserted.

In the User Key menu and any configuration menu, insert character acts the same as in Format mode for unprotected and transmit-only fields.

From the keyboard, you enable and disable the insert character editing function using the [Insert char] key. When enabled, the characters "`Ins Char`" are displayed in the status line at the bottom of the screen.

From a program executing in a host computer, you enable and disable the insert character editing function using the following escape sequence:

ENABLE: `Ec Q`

DISABLE: `Ec R`

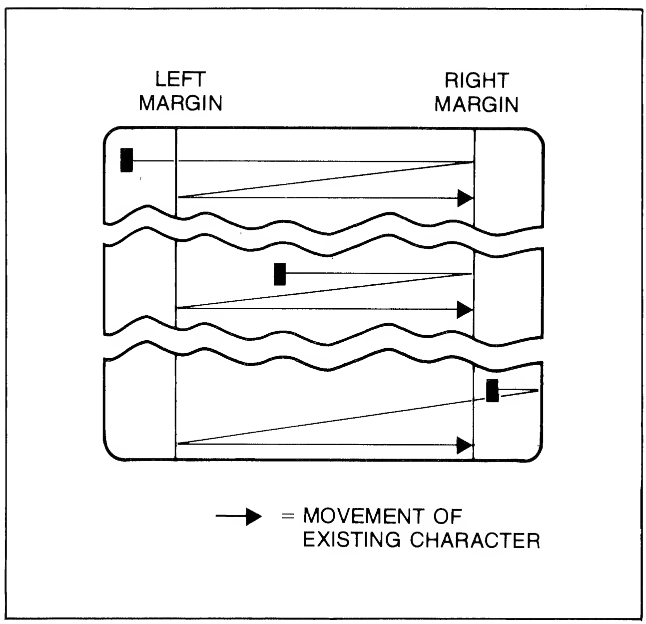
Insert Character with Wraparound

This edit function works the same as the insert character function except that characters forced beyond the right margin are not lost. When the rightmost non-blank character reaches the right margin any characters that are forced over the right margin move into (are inserted in) the next lower line at the left margin. If the next lower line becomes filled, a blank line is then inserted above it and the character overflow from the line being edited spills into the new line. As with the insert character function the cursor moves one column to the right (along with the existing data) each time a character is inserted and it progresses from the right margin of one line to the left margin of the next lower line.

This edit function is meant to be used within that portion of display memory bounded by the left and right margins. If you position the cursor to the left of the left margin, the insert character with wraparound function works as described above. If you position the cursor beyond the right margin, however, the insert character function is performed without wraparound until the cursor reaches the right boundary and moves to the left margin of the next lower text line. At that point the insert character function proceeds with wraparound within the defined margins.

The movement of existing characters during an "insert character with wraparound" editing operation is illustrated in figure 4-4.

Figure 4-4. Character Insert With Wraparound



When Format mode is off, any unprotected, transmit-only, alternate character set, and/or video enhancement fields to the right of the cursor move to right with the displayable characters. If part of such a field is forced over the right margin and into the next lower line, the character positions left within the field on the current cursor line maintain their characteristics while those that are wrapped lose their field characteristics, but maintain their alternate character set and video enhancement characteristics.

If all of such a field is forced over the right margin and into the next lower line, the character positions within the entire field maintain their characteristics.

If the cursor is positioned within any such field, the insert character with wraparound function extends the range of the field by one position for each character inserted, unless the end of the field is wrapped to the next line; in which case, the field will stay the same length.

Block terminators, at or to the right of the cursor position, move to the right along with the displayable characters. Non-displaying terminators to the right of the cursor position also move to the right along with the displayable character. A non-displaying terminator at the cursor position, hovever, remains at that position.

When Format mode is on and the cursor is positioned in an unprotected or transmit-only field, this function is performed without wraparound and affects only those characters from the cursor position through the end of the current field. Block terminators and non-displaying terminators are treated the same as when Format mode is off. If the cursor is not within an unprotected or transmit-only field, it automatically moves to the first character position of the next subsequent unprotected field when the first chracter is inserted.

From the keyboard, you may enable and disable the insert character with wraparound editing functions by using the [Shift] and [Insert char] keys. When enabled the characters "Ins Wrap" appear in the terminal's status line.

From a program executing in a host computer you enable and disable the insert character with wraparound editing function using the following escape sequence:

ENABLE: E_C N

DISABLE: E_C R

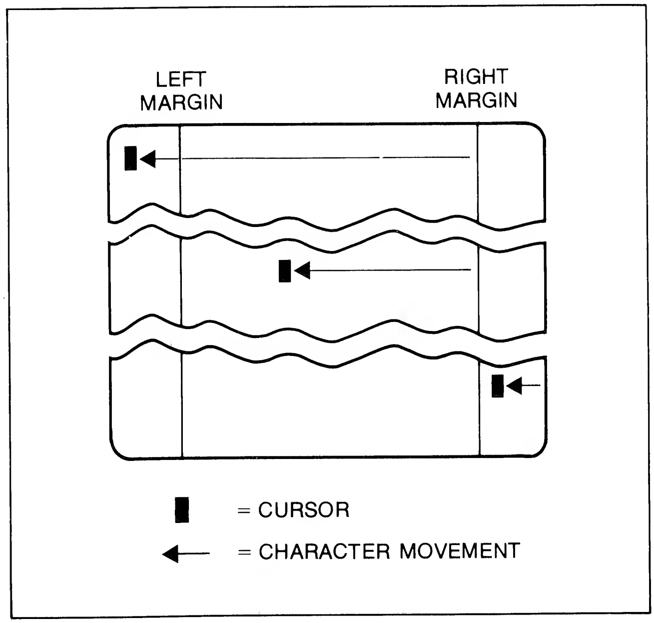
Delete Character

When you use the delete character edit function, the cursor remains stationary, the character at the cursor position is deleted, all characters between the cursor and the right margin move left one column, and a blank moves into the line from the right margin.

This edit function is meant to be used within the part of display memory bounded by the left and right margins. If you position the cursor to the left of the left margin, the delete character function works as described above. If you position the cursor beyond the right margin, however, the delete character function affects only those characters from the current cursor position through the right boundary of the screen.

The movement of existing characters during a "delete character" editing operation is illustrated in figure 4-5.

Figure 4-5. Character Delete with Margins



When Format mode is off, any unprotected, transmit-only, alternate character set, and/or video enhancement fields to the right of the cursor move to the left with the displayable characters. If the cursor is positioned within any such field, the delete character function shortens the range of the field by one position for each character deleted (you cannot, however, delete the end-of-field marker in unprotected or transmit-only fields). Deleting the first character position of an unprotected field changes the rest of the field to protected. Deleting characters at the start of, or within, a video enhancement and/or alternate character set field does not alter the characteristics of the rest of the field. Block terminators and non-displaying terminators to the right of the

cursor move to the left along with the displayable characters and are deleted if they are at the cursor position when this function is executed.

When Format mode is on and the cursor is positioned within an unprotected or transmit-only field, this function affects only those characters from the cursor position through the end of the current field. If the field definition also includes a video enhancement and/or an alternate character set, those characteristics are not altered by the delete character function. Block terminators and non-displaying terminators are treated the same as when Format mode is off. If the cursor is not within an unprotected or transmit-only field, the delete character function has no effect.

In the User Key menu and any configuration menu, delete character acts the same as for unprotected and transmit-only fields in Format mode.

From the keyboard, each time you press the [Delete char] key the terminal deletes one character. If you hold the key down, the terminal continues to delete characters until either the key is released or there are no non-blank characters between the cursor position and the right margin. In the latter case, pressing or continuing to hold down this key has no further effect.

From a program executing in a host computer, you delete the character at the current cursor position using the following escape sequence:

```
Ec P
```

Delete Character with Wraparound

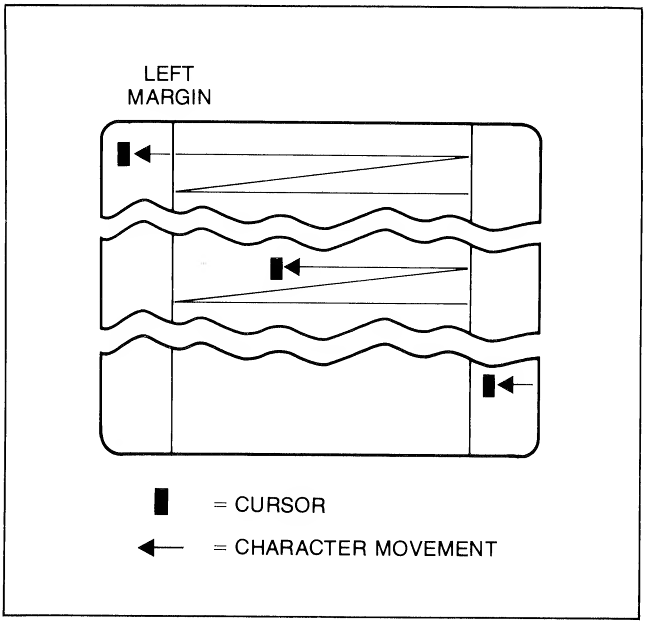
When you use the delete character with wraparound edit function, the cursor remains stationary, the character at the cursor position is deleted, all characters between the cursor and the right margin roll left one column, and one character rolls from the left margin of the next lower text line into the current line from the right margin. As a character rolls in from the next lower line, the remaining characters in that line roll one column to the left and a blank rolls in from the right margin.

The delete character with wraparound edit function affects only the line containing the cursor and the next lower text line.

This edit function is meant to be used in the part of display memory bounded by the left and right margins. If you position the cursor to the left of the left margin, the delete character with wraparound function works as described above. If you position the cursor beyond the right margin, however, the delete character function is performed without wraparound and it affects only those characters from the cursor position through the right boundary of display memory.

The movement of existing characters during a "delete character with wraparound" editing operation is illustrated in figure 4-6.

Figure 4-6. Delete Character with Wraparound



When Format mode is off, any unprotected, transmit-only, alternate character set, and/or video enhancement fields to the right of or in the line below the cursor move to the left with the displayable characters. If part or all of such a field moves into the line containing the cursor the character positions that have been wrapped maintain their field, character set, and video enhancement characteristics. Those characters that were not wrapped will lose only their field characteristics. If the cursor is positioned within any such field, the delete character with wraparound function shortens the range of the field by one position for each character deleted (you cannot, however, delete the end-of-field marker in unprotected or transmit-only fields). Deleting the

first character position of an unprotected or transmit-only field changes the rest of the field to protected. Deleting characters at the start of or within a video enhancement and/or alternate character set field does not alter the characteristics of the rest of the field. Block terminators and non-displaying terminators to the right of the cursor move to the left along with the displayable characters and are deleted if they are at the cursor position when this function is executed.

From the keyboard, each time you press the [Shift] and [Delete char] keys, the terminal deletes one character with wraparound. If you hold these keys down, the terminal continues to delete characters with wraparound until either the keys are released or until there are no non-blank characters remaining between the cursor and the right margin of the next lower line.

From a program executing in a host computer, you delete the character at the current cursor position with wraparound using the following escape sequence:

```
Ec O
```

Clear Display

When not in Format mode, pressing the [Clear display] key deletes all displaying and non-displaying characters from the current cursor position through the end of display memory.

By pressing the [▽] key followed by the [Clear display] key, you can clear all of alphanumeric display memory. This can also be done by pressing the [Shift] and [Clear display] keys together.

In Format mode, pressing the [Clear display] key deletes all unprotected displaying and non-displaying characters (except protected video enhancement escape sequences) from the current cursor position through the end of display memory.

The [Clear display] key is disabled when the User Key menu or a configuration menu is displayed.

The display can also be cleared by pressing the ⏷ key followed by the [Clear display] key.

To perform this function programmatically, use the following escape sequence:

```
Ec J
```

Clear Line

When Format mode is off, pressing the [Clear line] key deletes all displaying and non-displaying characters from the current cursor position through the end of the current line. Pressing [Shift] and [Clear line] together deletes the entire line.

When Format mode is on and the cursor is positioned within an unprotected field, pressing the [Clear line] key deletes all displaying and non-displaying characters (except video enhancement escape sequences) from the current cursor position through the end of the current field. If the cursor is not within an unprotected field, the [Clear line] key has no effect.

In the User Key menu and any configuration menu, clear line acts the same as for unprotected and transmit-only fields in Format mode.

To perform this function programmatically, use the following escape sequence:

```
Ec K
```

Extended Character Use

The terminal includes an extended character set containing special characters used in the international languages supported by the terminal. This set of characters is accessed using the [Extend char] keys. Refer to Appendix D, Keyboards and Character Sets, for detailed information.

Oversize Characters

Using an escape sequence, it is possible specify certain characteristics to apply to characters existing on a line, or to any subsequently entered on the line. These characteristics are:

- Double-wide characters.
- The top half of double-high characters.
- The bottom half of double-high characters.
- The top half of double-high, double-wide characters.
- The bottom half of double-high, double-wide characters.

It is also possible to convert a line with the above-listed characteristics back to a standard line using the same escape sequence.

When a line of standard-size characters is converted to double wide, twice the original number of character cells are required to represent all the characters on the line. If the number of characters is too great for the width of the line, the excess characters on the right end will not be displayed, although they will be retained in memory and can be displayed by returning the characters to their normal size.

To produce a line of double-high characters, two sequential lines must be used, one for the top half and one for the bottom half of the characters. The same characters must be entered in each line. When entered in the upper line, the top half of the characters will be displayed; when entered in the lower line, the bottom half of the characters will be displayed, to complete the double-high characters.

The escape sequence is as follows:

```
Ec&k<code>S
```

where <code> has the following selections:

<code>	SELECTION
-1	Display only the top half of the characters, double-high.
-2	Display only the bottom half of the characters, double-high.
-3	Display only the top half of the characters, double-high, double-wide.
-4	Display only the bottom half of the characters, double-high, double-wide.
-5	Display standard size characters.
-6	Display standard-size high, double-wide characters.

Display Enhancements

The terminal includes as a standard feature the display enhancements listed below:

■ Inverse Video	Black characters are displayed against a white background.
■ Underline Video	Characters are underscored.
■ Blink Video	Characters blink on and off.
■ Half Bright	Characters (or background, for inverse video) are displayed at half intensity.
■ Security Video	Character display is suppressed this enhancement is used in conjunction with fields in which passwords, or similar security-sensitive data, must be entered through the keyboard).
■ Inverse Background	Inverts the background and foreground, so that displayed characters are black on a white background. With this display enhancement enabled, the effect of the inverse video function is reversed, so as to produce the normal background/foreground relationship.

You use the first five enhancements on a field basis. They may be used separately or in any combination. When used, they cause control bits to be set in display memory. If the content of display memory is subsequently transmitted in Block mode to a host computer, these control bits are translated into escape sequences which are transmitted along with the displayable text characters (the same is true if the "`Esc Xfer(N)`" field on the Terminal Configuration menu is set to "`Yes`" and you are copying the content of display memory to an external printer). The inverse

background enhancement, on the other hand, is entirely a local function in that it affects the appearance of the display screen, but does not set any control bits in display memory.

Using Display Enhancements

From the Keyboard

From the keyboard, you enable and disable the various video enhancements using the function keys (except for inverse background, which is enabled and disabled on the Global Configuration menu). To do so you must first enable the video enhancement keys as follows:

System, enhance video

To cause a particular string of text characters to be displayed using one or more of the enhancements, do as follows:

1. Enable the desired enhancement(s) by pressing the associated function key. When an enhancement is enabled, an asterisk appears in the associated key display.
2. Position the cursor at the first character in the string.
3. Press SET ENHNCMNT. The selected enhancements take effect immediately. The enhancements begin at the cursor position and continue either through the right boundary of display memory, through the rightmost character on the line, to a character set change, or to the next column in which another display enhancement begins. When you press SET ENHNCMNT, the asterisks automatically disappear from the function key display (all enhancements are disabled until you once again enable them).
4. Position the cursor at the column immediately to the right of the final character in the string.

5. Press the SET ENHNCMNT key. The enhancements disappear from the cursor position either through the right boundary of display memory, through the rightmost character on the line, to a character set change, or to the next column in which another display enhancement begins. You have actually enabled "no enhancements" which is recorded in display memory as a control bit pattern that will be translated into an escape sequence (`Ec&d@`) if the content of display memory is tranmitted to a host computer in Block mode.

From a Program

From a program executing in a host computer you enable and disable the video enhancements by embedding escape sequences in the data. The general form of the escape sequence is as follows:

```
Ec&d <enhancement code>
```

where <enhancement code> is an @, s, or S, or one of the uppercase letters A through O specifying the desired enhancement(s) as follows:

CODE	BLINKING	INVERSE VIDEO	UNDERLINE	HALF BRIGHT
@				
A	X			
B		X		
C	X	X		
D			X	
E	X		X	
F		X	X	
G	X	X	X	
H				X
I	X			X
J		X		X
K	X	X		X
L			X	X
M	X		X	X
N		X	X	X
O	X	X	X	X

To enable and disable the security field enhancement, use the character "s" or "S" as an <enhancement code>. For example, consider the following escape sequences:

Ec&dsB	Enable both the security field and inverse video display enhancements (and disable any other existing enhancements at the current cursor position).
Ec&dS	Enable the security field enhancement by itself (and disable any other existing enhancement at the current cursor position).

Note that "s" or "S" can be used in conjunction with other enhancement codes, but it must precede the other enhancement.

Using Color In Text

The following information applies to the HP 2397A terminal only.

Each character cell in the window can be assigned two colors: the background color, which is the color for the whole cell, and the foreground color, which is the color of the character occupying the cell and overlaying the background color.

The background and foreground colors for a character cell are called a color pair.

Eight color pairs, identified by the numbers 0 through 7, are defined for the terminal. You can use the provided definitions or replace any or all provided pairs with your own selections. The provided pairs are shown below:

COLOR PAIR	FOREGROUND COLOR	BACKGROUND COLOR
0	White	Black
1	Red	Black
2	Green	Black
3	Yellow	Black
4	Blue	Black
5	Magenta	Black
6	Cyan	Black
7	Black	Yellow

To use your own selection of a color pair for creating text, you must:

- Define the color pair (Select the two colors).
- Select the color pair (select it as the one for use).

Defining a Color Pair

To select the colors for a color pair, three steps must be performed:

- Select the method of color identification, RGB or HSL (RGB is the default selection which is in effect after a power on, a hard reset, or a change made by the user).
- Identify the color pair (one digit, 0 through 7) for which the colors are to be selected. (The default selection is pair 0.)
- Select the colors for the color pair.

The elements used to identify a color (red, green, and blue or hue, saturation, and luminosity) are selected using numerical values in the range 0.00 to 1.00, varying in increments of 0.01. This range of values can be subdivided into smaller ranges, each of which represents a color band. For the RGB method of color identification, the numerical range for each color band is listed in table 4-1. Table 4-2 lists the ranges for the HSL method.

Table 4-1. RGB Color/Numerical Range Relationships

COLOR SELECTED	RED	GREEN	BLUE
Black	<0.25 or <N/2	<0.25 or <N/2	<0.25 or <N/2
White	>=0.25 and >=N/2	>=0.25 and >=N/2	>=0.25 and >=N/2
Yellow	>=0.25 and >=N/2	>=0.25 and >=N/2	<0.25 or <N/2
Green	<0.25 or <N/2	>=0.25 and >=N/2	<0.25 or <N/2
Cyan	<0.25 or <N/2	>=0.25 and >=N/2	>=0.25 and >=N/2
Blue	<0.25 or <N/2	<0.25 or <N/2	>=0.25 and >=N/2
Magenta	>=0.25 and >=N/2	<0.25 or <N/2	>=0.25 and >=N/2
Red	>=0.25 and >=N/2	<0.25 or <N/2	<0.25 or <N/2

Note: N represents the largest value of the three color components (red, green, and blue).

Table 4-2. HSL Color/Numerical Range Relationships

COLOR SELECTED	HUE	SATURATION	LUMINOSITY
Black	Don't care	Don't care	<0.25
White	Don't care	< 0.25	>=0.25
Red	0.00–0.08	>=0.25	>=0.25
Yellow	0.09–0.24	>=0.25	>=0.25
Green	0.25–0.41	>=0.25	>=0.25
Cyan	0.42–0.58	>=0.25	>=0.25
Blue	0.59–0.74	>=0.25	>=0.25
Magenta	0.75–0.91	>=0.25	>=0.25
Red	0.92–1.00	>=0.25	>=0.25

The escape sequence for selecting the colors for a color pair is as follows:

```
Ec&v <method>m <fclr1>a <fclr2>b <fclr3>c
               <bclr1>x <bclr2>y <bclr3>z
               <pair>i
```

(Spaces used in the escape sequence for clarity.)

where:

<method> 0 = RGB
1 = HSL
Default=0

<fclr1> Foreground value for red or hue (table 4-1 or 4-2). The default value is 0.

`<fclr2>` Foreground value for green or saturation table 4-1 or 4-2). The default value is 0.

`<fclr3>` Foreground value for blue or luminosity table 4-1 or 4-2). The default value is 0.

`<bclr1>` Background value for red or hue (table 4-1 or 4-2). The default value is 0.

`<bclr2>` Background value for green or saturation table 4-1 or 4-2). The default value is 0.

`<bclr3>` Background value for blue or luminosity table 4-1 or 4-2). The default value is 0.

`<pair>` Identity (one digit, 0–7) of the color pair for which colors are to be selected. The default pair is pair 0.

The escape sequence can contain any or all of the above parameters. If any of the parameters are missing from the escape sequence, they are set to 0.

Multiple color pairs can be defined in the same escape sequence.

If the `<pair>` is identified in the escape sequence, it must follow the background and foreground values.

Selecting a Color Pair

The default color pair is 0 (white foreground and black background), so after power on or a hard reset this is the color pair used. If one of the eight system-provided color-pair definitions is not to be used, you must define your own color pair, as described previously, then select it.

The selected color pair takes effect at the current cursor position and continues through the rightmost character on the line, or to the next column on the same line in which another color pair has been selected previously.

A color pair is selected using the following escape sequence:

```
Ec&v<pair>s
```

where:

`<pair>` is the identification number (0–7) assigned to the color pair when it was defined.

Sample Color Definition Values

Table 4-3 lists representative color component values for the basic colors for both the RGB and HSL color identification methods.

Table 4-3. Sample Color Definition Values

COLOR	R	G	B	H	S	L
Black	0	0	0	X	X	0
Blue	0	0	1	0.66	1	1
Green	0	1	0	0.33	1	1
Cyan	0	1	1	0.5	1	1
Red	1	0	0	1	1	1
Magenta	1	0	1	0.83	1	1
Yellow	1	1	0	0.16	1	1
White	1	1	1	X	0	1

X = don't care (may be any value between 0 and 1).

Example

In the following example, three color pairs (4, 5, and 6) are defined, then each is used to display three letters of the word "`chameleon`" on the screen. (Values for color definition are selected from table 4-3).

Step 1. Use the HSL method to define color pair 4 as red characters on a blue background by typing in the following escape sequence (spaces used in the escape sequence for clarity):

```
Ec&v 1m 1a 1b 1c .66x 1y 1z 4I
```

Step 2. Use the RGB method to define color pair 5 as green characters on a yellow background by typing in the following escape sequence (if the default value, 0, is to be used for one of the foreground or background color components, the parameter need not be included in the escape sequence):

```
Ec&v 0m 1b 1x 1y 5I
```

Step 3. Use the HSL method to define color pair 6 as blue characters on a white background by typing in the following escape sequence:

```
Ec&v 1m .66a 1b 1c 1z 6I
```

Step 4. Select color pair 4 for use by typing in the following escape sequence:

```
Ec&v 4S
```

Step 5. Type in the letters "`Cha`".

Step 6. Select color pair 5 by typing in the following escape sequence:

```
Ec&v 5S
```

Step 7. Type in the letters "`mel`".

Step 8. Select color pair 6 by typing in the following escape sequence:

```
Ec&v 6S
```

Step 9. Type in the letters "`eon`".

Determining the Status of a Color Pair

You can determine the identity, method of color selection, and the foreground and background color values of the any color pair using an escape code. Refer to Section 9, Status, for details.

Display Enhancements for Color Terminals

For color terminals, several display enhancements have unique features, as listed below. Except for these features, display enhancements for color terminals are the same as for monochrome terminals.

Note The security video enhancement is not incorporated in color terminals.

■ Half Bright	For color terminals, halfbright display enhancement substitutes color pair 3 in place of reduced intensity. Both foreground and background colors are full intensity.
■ Inverse Background	Inverts the background and foreground colors for each character.

Coloring the Function Key Labels

Using an escape sequence, each function key label can be assigned the colors of any one of the eight color pairs. The escape sequence used is the one for assigning labels, character strings, and enhancements to the function key. Refer to Section 11, User-Definable Function Keys for further information.

5 Terminal Control

Introduction

This section describes the modes in which the terminal can operate and controlling the terminal from the keyboard.

The terminal can operate in one of three major modes: ANSI mode, EM52 mode, and HP mode. The mode is selected on the Global Configuration menu.

ANSI Mode

ANSI mode enables use of the terminal with a computer system which uses ANSI protocol. In ANSI mode, ANSI functions are assigned to some of the keyboard keys in place of their normal function in HP mode. Also, the terminal is capable of executing a subset of ANSI terminal escape sequences, as well as some HP mode escape sequences.

In ANSI mode, the terminal doesn't respond to VT52 escape sequences, although it does respond to VT100 escape sequences. Refer to Appendix B for further details.

Note As used in this manual, The acronym "ANSI" always refers to the standard, ANSI X3.64, and not to the ANSI organization. Also, use of the term in no way implies approval of this product by the ANSI organization.

EM52 Mode

In EM52 mode, the terminal emulates a DEC VT52®-compatible terminal. As in ANSI mode, some of the keyboard key functions are changed. The terminal responds to the VT52® terminal escape sequences listed in Appendix A, and selected HP mode escape sequences. For further details on VT52 mode, refer to Appendix B.

HP Mode

All information in this manual, including the remainder of this section, but excepting Appendix B, applies to HP mode. In HP mode, the terminal does not respond to ANSI or VT52 mode escape sequences.

Selecting Modes in HP Mode

Within HP mode, most primary terminal modes are accessed through the Modes set of function keys. The Modes keys are accessed by pressing the [System] and [modes] keys, in succession. (If the terminal is in User Keys mode, the [System] key must be pressed twice in succession.)

The Modes set of function keys are shown below.

LINE MODIFY	MODIFY ALL	BLOCK MODE	REMOTE MODE	SMOOTH SCROLL	MEMORY LOCK	DISPLAY FUNCTNS	AUTO LF

These keys act as toggle switches; they alternately enable and disable the designated mode. When a mode is enabled, an asterisk is displayed in the label.

Remote/Local Modes

When a communications link exists between the terminal and a remote host computer, the terminal is in one of of the following two modes:

- Remote Mode. In this mode, when you press an alphanumeric key the associated ASCII code is transmitted to the host computer.
- Local Mode. In this mode, when you press an alphanumeric key the associated character is displayed at the current cursor position on the screen (nothing is transmitted to the host computer).

From the keyboard, you switch the terminal back and forth between Local and Remote modes using the REMOTE MODE key.

From the keyboard or a user-definable key, you can switch the terminal from local to remote (and vice versa) using the following escape sequences:

Local:	`Ec &k0R`
Remote:	`Ec &k1R`

After a hard reset or turning off the power, the terminal reverts to Remote mode.

Character/Block Modes

When the terminal is connected on-line to a remote host computer, it operates in either of the following data transmission modes:

- Character Mode. In this mode, data is transmitted a character at a time as it is entered through the keyboard. ASCII control codes (such as <CR> and <LF>) are transmitted.

- Block Mode. In this mode, data is not transmitted at the time it is entered through the keyboard. Instead, you transmit an entire block of data by pressing the [Enter] key after typing and editing the data.

When the terminal is in Block mode, ASCII control codes (such as <CR> and <LF>) are acted upon locally but are not usually transmitted with the data block (refer to Section 9 for detailed information on Block mode data transfers).

From the keyboard, you enable and disable Block mode using the BLOCK MODE key.

From a program executing in a host computer, you enable and disable Block mode using the following escape sequences:

ENABLE: `Ec &k1B`

DISABLE: `Ec &k0B`

After a hard reset or turning off the power, the terminal reverts to Character mode.

The relationship between Block, Line, Page, and Format modes is described in Section 11, Block Data Transfers.

Line Modify Mode

When the terminal is in Remote mode and Character mode, and you are communicating interactively with a host computer, you may sometimes enter an erroneous character string to which the computer responds with an error message. If the character string is a lengthy one and the error consists of only a few characters, it is a nuisance to have to retype the entire string. In such a case, you may instead enable Line Modify mode (which temporarily switches the terminal to a special form of Block mode). You may then move the cursor to the erroneous line on the display and correct the character string. When the string is edited to your satisfaction, you retransmit the line to the host computer by pressing either the Return key or the Enter key.

Note that while Line Modify mode results in a block transmission, it is completely independent of the Block mode function described earlier in this section (you do not have to first enable Block mode). In fact, Line Modify mode

is a feature that was specifically designed for use when the terminal is operating in Character mode, and will function only in Character mode.

From the keyboard, you enable Line Modify mode using the LINE MODIFY key. Line Modify mode is automatically disabled when you press either [Return] or [Enter]. If you change your mind and wish to disable Line Modify mode before retransmitting the command string, press the LINE MODIFY key again and the terminal will return to normal Character mode.

For each line, the terminal remembers the position of the first character entered from the keyboard. Then when you re-transmit a line in modify mode, only the portion of the line entered from the keyboard is retransmitted; any prompt characters preceding the command string are ignored by the terminal. For more detailed information about this feature refer to the discussion of the "Start Column" field of the Terminal Configuration menu in Section 6. For information on handshaking and block-terminating characters, refer to Section 11.

Modify All Mode

When the terminal is in Character mode, you can enable Modify All mode, which switches the terminal to a special form of Block mode. Modify All mode is the same as Line Modify mode except that it is not disabled when you press [Return] or [Enter].

From the keyboard, you enable and disable Modify All mode using the MODIFY ALL key.

From a program executing in a host computer, you enable and disable Modify All mode using the following escape sequences:

ENABLE: `Ec &k1M`

DISABLE: `Ec &k0M`

After a hard reset or turning off the power, the terminal reverts to a modify modes off condition.

For information on handshaking and block-terminating characters associated with transfer of the data block, refer to Section 10.

Auto Line Feed Mode

When Auto Line Feed mode is enabled, an ASCII line feed control code is automatically appended to each ASCII carriage return control code generated through the keyboard. That is, every <CR> code generated through the keyboard becomes a <CR><LF>.

ASCII carriage return control codes can be generated through the keyboard in any of the following ways:

- By pressing the [Return] key, provided that a <CR> code is included in the key definition.
- By simultaneously pressing the [CTRL] and [M] keys.
- By pressing any of the user keys ([f1] through [f8]), provided that a <CR> code is included in the particular key definition.
- By pressing the [Enter] key when the terminal is in Block mode, Line Modify mode, or Modify All mode (in these cases a <CR> code is transmitted as the line terminator).

From the keyboard, you enable and disable Auto Line Feed mode using the AUTO LF key.

From a program executing in a host computer, you enable and disable Auto Line Feed mode using the following escape sequences:

ENABLE: `Ec &k1A`

DISABLE: `Ec &k0A`

After a hard reset or turning off the power, the terminal reverts to the Auto Line Feed mode off condition.

Display Functions Mode

When Display Functions mode is enabled the terminal operates as follows:

- In Local mode, it displays ASCII control codes and escape sequences but does not execute them. For example, if you press the ◄ key the terminal displays `Ec D` on the screen but does not perform the "cursor left" function.

- In Remote mode, it transmits ASCII control codes and escape sequences but does not execute them locally. For example, if you press the Shift ▲ keys the terminal transmits an `Ec S` but does not perform the "roll up" function.

There are two exceptions to the foregoing descriptions:

1. An `Ec Y`, which enables Display Functions mode, is executed ,but is not transmitted or displayed; `Ec Z`, which disables Display Functions mode, is both displayed and executed, but not transmitted.
2. A <CR> (or <CR><LF> if Auto Line Feed mode is enabled) is executed in addition to being transmitted and/or displayed.

From the keyboard, you enable and disable Display Functions mode using the DISPLAY FUNCTNS key.

From a program executing in a host computer, you enable and disable Display Functions mode using the following escape sequences:

ENABLE: `Ec Y`

DISABLE: `E1 Z`

Note

There is interaction between Display Functions and the "`XmitFnctn(A)`" field of the Terminal Configuration menu. If `XmitFnctn(A)` is on, the DISPLAY FUNCTNS key transmits `Ec Y`. However, the DISPLAY FUNCTNS key will not transmit `Ec Z` when pressed again.

Once enabled, Display Functions mode remains enabled until explicitly disabled, until a soft or hard reset is performed, or until the power is turned off.

Memory Lock Mode

Memory lock mode provides two separate functions: overflow protect and display lock.

Overflow Protect. If you home the cursor and then enable Memory Lock mode, the workspace becomes "protected" so that no data can be lost off the top. In such a case, when you have used all available lines in the workspace, any attempt to use more memory is rejected with an audible "beep" and the message "`MEMORY FULL`" is displayed. You may, however, use the cursor control keys to go back and alter any of the existing data. To continue entering new data, merely disable Memory Lock mode and reposition the cursor immediately below the last line. Before doing so you may wish to enable "log top" data logging (described in Section 7) so that data that is then forced off the top of the workspace will be retained in printed form.

Display Lock. If you position the cursor below the top line of the screen and then enable Memory Lock mode, the lines above the cursor line become "locked" on the screen. As the screen becomes full, the locked lines remain on the screen while subsequent lines roll past the locked rows. This allows you to retain column headings or instructions on the screen as you continue to enter new data. It also provides a useful means of changing the sequence of text blocks as follows:

a. Press [▽], [Clear display], and then type the following data:

```
3. This is paragraph 3.
   It should be the third one.

1. This is paragraph 1.
   It should be the first one.

2. This is paragraph 2.
   It should be the second one.

4. This is paragraph 4.
   It should be the last one.
```

b. Position the cursor in the first line of paragraph 1.

c. Enable Memory Lock mode.

d. Use the [Shift] [▲] keys until the first line of paragraph 4 is in the same line as the cursor.

e. Disable Memory Lock mode and home the cursor. The display should appear as follows:

```
1. This is paragraph 1.
   It should be the first one.

2. This is paragraph 2.
   It should be the second one.

3. This is paragraph 3.
   It should be the third one.

4. This is paragraph 4.
   It should be the last one.
```

From the keyboard, you enable and disable Memory Lock mode using the MEMORY LOCK key. The rows above the line containing the cursor are locked.

Normal editing can be performed within the locked rows; that is, the rows are locked by row number only, so if lines are inserted among the locked rows, they become locked but the total number of locked rows does not increase.

From a program executing in a host computer, you enable and disable Memory Lock mode using the following escape sequences:

ENABLE: `Ec l`

DISABLE: `Ec m`

Once enabled, Memory Lock mode remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Smooth Scroll Mode

Smooth Scroll mode is selected with the SMOOTH SCROLL function key. As implied in the name, lines are scrolled smoothly on or off the screen in Smooth Scroll mode, so that you can watch them appear or disappear. When the terminal is not in Smooth Scroll mode, the top or bottom line appears or disappears immediately, with no transition apparent to the eye.

Caution

Use of Smooth Scroll mode, while in Remote mode, may result in lost data, unless a type of handshaking is used in which the terminal controls transmission of data it receives. To ensure against lost data, set the "EnqAck" field on the configuration menu in use to "Yes", for HP systems. For non-HP systems, set the "RecvPace" field to XON/XOFF.

Caps Lock Mode

When Caps Lock mode is enabled, the terminal generates only Teletype compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a–z) generate the codes for their uppercase equivalents, the {, |, and } keys generate the codes for [, \, and] (respectively), and the ` and ~ keys are ignored.

From the keyboard, you enable and disable Caps Lock mode using the "Caps Lock" field of the Terminal Configuration menu described in Section 6.

From a program executing in a host computer, you enable and disable Caps Lock mode using the following escape sequences:

ENABLE: `Ec &k1C`

DISABLE: `Ec &k0C`

At any given time the current state (enabled/disabled) of Caps Lock mode is reflected in the "`Caps Lock`" field of the terminal configuration menu. When you enable or disable the mode by altering the menu field from the keyboard and then pressing the SAVE CONFIG key, you alter both the active and non-volatile memory versions of that field. When you enable or disable the mode using the escape sequence, however, you only change the active value of the "`Caps Lock`" field in the terminal configuration menu.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the "`Caps Lock`" field in the Terminal Configuration menu in non-volatile memory.

Format Mode

The terminal has a Format mode in which custom-designed forms can be displayed on the screen and used for data entry. These forms contain unprotected fields, in which data is entered, and protected fields, which contain the unchanging printed information (such as titles or names of fields). Refer to Section 12 for more information on data entry forms.

Normal procedure is to design the form, display it on the screen, then enter Format mode to enter the data and transmit it to the computer.

When Format mode is enabled, the terminal operator may enter data into unprotected fields only. If the operator positions the cursor in a protected area and then attempts to type data, the cursor automatically moves to the start of the next subsequent unprotected field before the terminal accepts the data.

When all data is entered, it is transmitted to the host computer with the [Enter] key.

From the keyboard, you enable and disable Format mode using the FORMAT MODE function key, which is available in the Define Fields set of function keys.

From a program executing in a host computer, you enable and disable Format mode using the following escape sequences:

ENABLE: `Ec W`

DISABLE: `Ec X`

Once Format mode is enabled, it remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Status Line

The lowest line on the display (line 27) is the status line. This line indicates the status of a number of terminal modes. Eight of the fields for the mode indications are located under the function keys, with three fields located beneath the area between the function keys, as shown below.

KB Lockd	Ext Char	Tab=Spac or L1L2L3L4	Num Pad or Grph Pad	* HP * or ANSI or EM52	CAPS	Ins Char or Ins Wrap	TouchOff or Touch On	STOP

Each indication has two states: present or not present. For example if the keyboard is locked, the "KB Locked" indication will be present beneath the leftmost function key; otherwise the field will be blank.

The meaning of each indication is as follows:

FIELD	DESCRIPTION
KB Lockd	Indicates whether or not the keyboard is locked.
Ext Char	Indicates whether the terminal is in Extended Character mode.Refer to Appendix D, Keyboards and Character Sets for information on Extended Character mode.
Tab=Spac	Indicates whether the terminal will generate a space character or a tab (HT) character when the [Tab] key is pressed. This selection is made on the Terminal Configuration menu, field "Tab=Spaces". This indicator is active only in HP mode.
L1L2L3L4	The L1, L2, L3, and L4 displays simulate four flags on an ANSI-compatible terminal. They appear only in ANSI mode.
Keypad	Num Pad - The numeric/graphics pad is in indicator Numeric mode. The keypad function is switched between numeric and graphics by pressing the [Shift] and numeric/graphics keypad [-] keys together. Grph Pad -The numeric/graphics pad is in Graphics mode.
Left asterisk	Datacomm port 1 is in Transmit mode.

FIELD		DESCRIPTION
Mode indicator	HP -	Terminal operates in its primary role as a Hewlett-Packard terminal.
	ANSI -	Terminal operates as an ANSI-compatible terminal.
	EM52 -	Terminal emulates a DEC VT52(R)-compatible terminal.
Right asterisk	Datacomm port 2 (on terminals with option 092 only) is in Transmit mode.	
CAPS	Terminal is in Caps Lock mode.	
Character insert mode	Ins Char -	Terminal is in Insert Character mode (without wraparound).
	Ins Wrap -	Terminal is in Insert Character mode (with wraparound).
TouchOff	Touch screen functions are disabled.	
Touch On	Touch screen functions are enabled.	
STOP	The [Stop] key has been pressed, stopping datacomm reception.	

Keyboard Data Entry

Data can be sent to the computer from the keyboard with the [Enter] key or by typing in the escape sequence `Ec d`. Refer to Section 10 for further information.

Keyboard Controls

Enable/Disable Keyboard

You can enable and disable the terminal's keyboard by executing escape sequences. When the keyboard is disabled all keys except the following are ignored:

[Shift], [CTRL], [Reset], and [Break]

The escape sequences for enabling and disabling the keyboard are as follows:

ENABLE: `Ec b`

DISABLE: `Ec c`

Once disabled, the keyboard remains disabled until explicitly enabled, until a soft or hard reset is performed, or until the power is turned off.

Soft Reset

A soft reset does the following:

1. Rings the terminal's bell.
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled).
4. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
5. Disables Display Functions mode (if enabled).
6. Halts any datacomm transfers currently in progress, clears the datacomm buffers.
7. Cancels any pending status requests.
8. Reinitializes all input devices in the HP-HIL link.

The data on the screen, all terminal operating modes (except Display Functions mode), and all active configuration parameters are unchanged.

From the keyboard, you perform a soft reset by pressing the Shift and Reset keys, together.

From a program executing in a host computer,you perform a soft reset using the following escape sequence:

```
Ec g
```

Hard Reset

For the currently active terminal personality, a hard reset has the same effect as turning the terminal's power off and then back on except that the power-on self-test is not performed.

More specifically, a hard reset does the following:

1. Rings the terminal's bell.
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled).
4. Clears both alphanumeric and graphics memories.
5. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
6. Halts any datacomm transfers currently in progress, clears the datacomm buffer, and reinitializes the datacomm port according to the appropriate power-on datacomm configuration parameters.
7. Resets the terminal configuration menu parameters to their power-on values.
8. Reset the user-programmable function keys to their default settings.

9. Resets certain operating modes and parameters as follows:

   Disables Display Functions mode, Caps Lock mode, Report mode, Metric mode, and data logging.

   Resets the left margin to the leftmost screen column.

   Resets the right margin to the rightmost screen column.

   Turns off the "insert character" edit function.

   Clears graphics memory.

From the keyboard, you perform a hard reset by simultaneously pressing the [Shift], [CTRL], and [Reset] keys.

From a program executing in a host computer, you perform a hard reset using the following escape sequence:

```
Ec E
```

Select Key

The [Select] key gives the keyboard some of the capabilities of the touchscreen, tablet, and mouse. The [Select] key is considered a touchscreen function. It is turned on and off when the touchscreen is turned on and off. The following escape sequence toggles the touchscreen on and off:

```
Ec-z 4N
```

The on or off condition of the touchscreen is shown in the status line.

Touchscreen Simulation. Pressing the [Select] key in Alphanumeric mode, performs a "pick" function, as though the terminal was equipped with a touchscreen. This action occurs even if the terminal is equipped with a touchscreen.

Tablet/Mouse Simulation. In Graphics mode, the [Select] key can simulate the "pen down" and "pen up" operations of a tablet or mouse. Pressing [Shift] and [Select] keys together toggles between the "pen up" and "pen down" conditions. Pressing [CTRL] and [Select] together, or [Shift], [CTRL], and [Select] together has the same effect.

Break

Pressing the [Break] key serves as a "break" signal to interrupt terminal/computer communications.

Bell

The terminal contains a speaker for sounding an audible tone in response to the ASCII bell control code and for alerting the terminal operator when certain error conditions occur.

From the keyboard, you generate the Bell code by simultaneously pressing the [CTRL] and [G] keys.

From a program executing in a host computer, you trigger the bell tone by transmitting an ASCII Bell control code (decimal 7).

Stopping and Starting Computer Data Transmission

Transmission of data from the host computer to the terminal can be stopped and restarted from the keyboard by pressing the [Stop] key. This key toggles between stopping and restarting data transmission. It does so by sending a `DC3` (`XOFF` - stop) or `DC1` (`XON` - resume).

Simultaneously pressing [CTRL] and [Q] also stops transmission by sending a `DC3`; [CTRL] [S] resumes transmission by sending a `DC1`.

6 Configuring the Terminal

Introduction

Configuration parameters may be changed from the keyboard via configuration menus or, in some cases, programmatically, by escape sequences.

A menu is an arrangement of configuration parameters displayed on the screen. Each parameter has an associated space for a value which you select. Many of the parameters have a system-defined list of values; for others, you must enter the value from the keyboard. For parameters with system-defined values, two function key labels are displayed with the menu to enable you to scroll forward NEXT CHOICE or backward PREVIOUS CHOICE through the list of values.

Nonvolatile Memory

Nonvolatile memory is a portion of terminal memory in which a set of values is preserved for all configuration menus and the f9–f12 User Key menu, whether or not power is applied to the terminal. The set stored is the one last stored by the user. If none has been stored by the user, the default set is stored.

When a menu is displayed, the values currently active are displayed. When the terminal is powered up, the set of values stored in nonvolatile memory becomes the active set.

Port 2 I/O Options

The terminal has three I/O options installable at port 2:

- Option 046—the HP-IB interface option.
- Option 092—the 25-pin RS232C interface option.
- Option 093—the 8-bit parallel Centonics interface option.

At power on, the terminal senses which option is installed in port 2 and will display the appropriate function key label in the "config keys" set of labels, so that the correct menu can be displayed.

Configuration from the Keyboard

The sequence for changing a set of configuration values is to display the menu, make the desired changes, and store the values in nonvolatile memory. The act of storing the values in nonvolatile memory also makes them the active set.

Some of the content of these menus may also be altered from a program executing on a host computer through the use of escape sequences. The changes made by the host computer are temporary and will be lost through hard reset or power down. That is, the changes are not saved in non-volatile memory.

Configuration Menus

Configuration requirements for the terminal and its ports are contained on eight menus, as listed below:

- Global Configuration menu
- Terminal Configuration menu
- Remote Datacomm Hardwired menu
- Remote Datacomm Modem menu
- External Serial Device menu
- External Parallel Device menu

- External HP-IB Device Configuration menu
- ANSI Configuration menu

Two additional menus, the User Keys menus, are used for defining the type, label, and definition of each of the function keys. Refer to "User Definable Function Keys" in Section 11 for further information on the User Key menu.

The terminal can display only one of the three external device menus (External Serial Device menu, External Parallel Device menu, and External HP-IB Device menu), depending on the option selected for the second port.

How To Display A Menu

A menu is selected for display using the function keys. When it is displayed, it will contain the active values for that menu. If no values have been stored for the menu, the default values will be displayed.

To display a menu, perform the following steps:

1. Press the [System] key twice, then the **config keys** key.
2. Press the appropriate keys, as listed in table 6-1. Pressing the **ext dev config** key will display the appropriate menu (External Serial Device, External Parallel Device, or External HP-IB Device) for the option installed in the terminal. A menu will be displayed only if an interface module, used for connecting an external device, is installed.

Table 6-1. Configuration Menu Selection Keys

MENU	KEYS
Global	global config
Terminal	terminal config
Remote Datacomm Hardwired	datacomm config config menus FULL DUP HRDWIRED
Remote Datacomm Modem	datacomm config config menus FULL DUP MODEM
Ext. Serial Dev.	ext dev config
Ext. Parallel Dev.	ext dev config
Ext. HP-IB Dev.	ext dev config
ANSI Config	ansi config

Modifying and Activating Configuration Values

Several function keys, displayed with configuration menus, can be used to display, select, change, and save configuration values. These keys and their functions are listed in table 6-2. A secondary set of function keys, the System Default set, are listed in table 6-3. The System Default set of keys enable selecting one of four sets of default configuration values, or the currently-active set, for the displayed menu.

Table 6-2. Function Keys Used for Menu Manipulation

LABEL	FUNCTION
SAVE CONFIG	Saves the displayed configuration parameters in nonvolatile memory, makes the set of parameters the active configuration set, and returns to normal operating mode with the System set of function key labels displayed.
NEXT CHOICE PREVIOUS CHOICE	Most of the fields on the menus have a list of acceptable values (some have only two). These keys scroll forward or backward through the list.
system defaults	Displays the "system default" set of function keys (table 6-3).
config menus	Displays the "config menus" set of function keys.
DISPLAY FUNCTNS	Alternately enables and disables Display Functions mode. When enabled, an asterisk is present in the label. Several menus contain fields for which entries must be made in Display Functions mode; for example, the "`Field Separator`" and "`Block Terminator`" fields on the Terminal Configuration menu. This key is used only for such entries on configuration menus, and does not affect the selection made with the DISPLAY FUNCTNS key in the system set of function keys.
config keys	Ends Configuration mode without saving the displayed values. Any changes made on the menu are lost. Returns to normal operating mode with the Configuration set of function key labels displayed.

Table 6-3. System Defaults Function Keys

LABEL	FUNCTION
config	Displays either the "datacomm config" or "terminal config" set of function keys, whichever set was displayed before this set.
POWER ON VALUES	Displays the values stored in nonvolatile memory, which become active at power-on time.
ACTIVE VALUES	Displays the values which are currently active for the configuration. (The active values might be different from the values stored in non-volatile memory.)
DEFAULT VALUES	Displays the default values for the displayed menu.
HP 1000 PT. PT.	Enters the default values, for connection to the HP 1000 computer, into every field of the displayed menu.
HP 3000 PT. PT.	Enters the default values, for connection to the HP 3000 computer, into every field of the displayed menu.

To change a selection on a menu, perform the following steps:

1. Place the cursor at the entry to be changed. This can be done using the Tab key or the cursor-positioning keys. The Tab key moves the cursor to the next or previous (when shifted) field each time the key is pressed.

2. If the choices are restricted to a system-defined list of selections (such a field is underlined, inverse video), use either the NEXT CHOICE or PREVIOUS CHOICE function key to cycle through the list of selections until the desired one is displayed.

If the choices are not restricted to a system-defined list (half bright, inverse video), enter the desired value from the keyboard.

3. To store the new menu values in nonvolatile memory after you have made all desired changes, press the **SAVE CONFIG** function key. This also makes the displayed values the active values.

To Return To Normal Operation

Pressing the **SAVE CONFIG** key returns the previous display contents to the display, saves the displayed configuration values in nonvolatile memory, and makes them the active values. If you wish to return the previous display contents to the screen without saving the displayed configuration values, you can press the [System] key or the [Shift] and [System] keys, simultaneously, or the **config keys** function key to do so. In this case, the previously active values remain the active values.

Global Configuration

To perform a global configuration, display the Global Configuration menu (figure 6-1), select the desired values, and save the selected values in nonvolatile memory, as described previously. The functions of the Global Configuration menu fields are listed in table 6-4. The default values for each menu field are shown in figure 6-1.

Figure 6-1. Global Configuration Menu

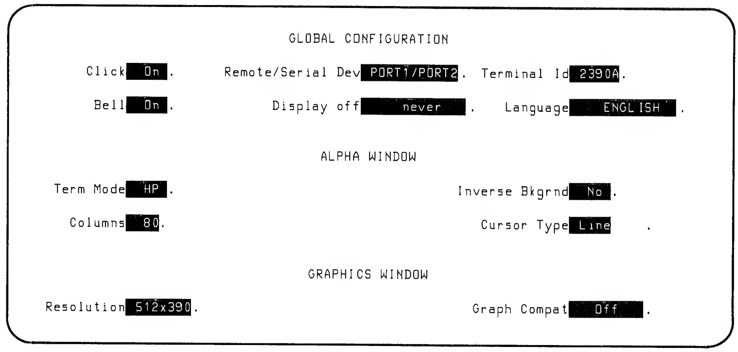


Table 6-4. Global Configuration Menu Fields

FIELD	FUNCTION
`Click`	Selects whether an audible click will occur when a keyboard key is pressed. `On`: Click selected `Off`: Click disabled Default=`On`
`Remote/Serial Dev`	Enables selection of either port 1 or port 2 as the host computer datacomm port, provided port 2 is an RS-232 port (Option 092). Otherwise, this field does not appear on the menu. **NOTE** If the entry in this field is changed and the SAVE CONFIG key is pressed, the terminal will perform a hard reset to switch to the new terminal mode, resulting in loss of all data in terminal memory. `PORT1/PORT2`: Port 1 is the host computer port; port 2 is the external device port. `PORT2/PORT1`: Port 2 is the host computer port; port 1 is the external device port. Default=`PORT1/PORT2`

Table 6-4. Global Configuration Menu Fields (continued)

FIELD	FUNCTION
`Terminal Id`	Enables selection of the self identification the terminal will send, on demand, to a program operating on the host computer. Values: Any alphanumeric string of five characters or less. Default=`2390A`
`Bell`	Enables selection of whether or not the bell will sound when the cursor enters the eighth character space left of the right margin during data entry in Non-Format mode. An error or pressing CTRL and G, simultaneously, will ring the bell regardless of the entry in this field. `On`: Bell enabled `Off`: Bell disabled Default=`On`
`Display Off`	Enables selection of several time durations for which the displayed data will continue to be displayed after the last HP-HIL datacomm activity. After the selected time interval, the screen will be blanked. Any HP-HIL datacomm device activity will automatically reset the timer and turn on the video without loss of data. `never`: Video displayed until power off. `5 min`: Video turned off 5 minutes after last activity. `10 min`: Video turned off 10 minutes after last activity. `15 min`: Video turned off 15 minutes after last activity. Default=`never`

(continued)

Table 6-4. Global Configuration Menu Fields (continued)

FIELD	FUNCTION
Language	Specifies in which language all function key labels and messages will be displayed. Values: ENGLISH ITALIANA DANSK NEDERLANDS DEUTSCH NORSK ESPANOL SUOMI FRANCAIS SVENSK Default=ENGLISH.
Term Mode	Selects one of three primary terminal operating modes. **NOTE** If the entry in this field is changed and the SAVE CONFIG key is pressed, the terminal will perform a hard reset to switch to the new terminal mode, resulting in loss of all data in terminal memory. HP: Terminal operates as a standard HP terminal, as described in this manual. ANSI: Terminal recognizes and executes a subset of the terminal escape sequences specified in the American National Standards Institute documents X3.14-1974 and X3.64-1979, in addition to some of the standard HP sequences. EM52: Terminal emulates a VT52 compatible terminal by recognizing and executing escape sequences written for a DEC VT52 terminal. Default=HP

(continued)

Table 6-4. Global Configuration Menu Fields (continued)

FIELD	FUNCTION
`Inverse Bkgrnd`	Selects whether the display screen background will be normal (black, displaying white letters) or inverse video (white, displaying black letters). Values: `Yes` (inverse) `No` (normal) Default=`No`
`Columns`	Enables selection of the width of the display memory workspace. If more than 80 columns are selected, horizontal scrolling must be used to view the entire workspace. Values: 80 to 160, in increments of 2. An odd number entry will be rounded up to the next even number. Default=`80`
`Cursor Type`	Enables selection of the cursor type. `Line`: blinking underline. `Box`: blinking box. Default=`Line`
`Resolution`	Enables selection of one of two levels of graphics display resolution. `512x390`: graphics display of 512 by 390 pixels selected. `640x400`: 640 by 400 pixels selected. Default=`512x390`

(continued)

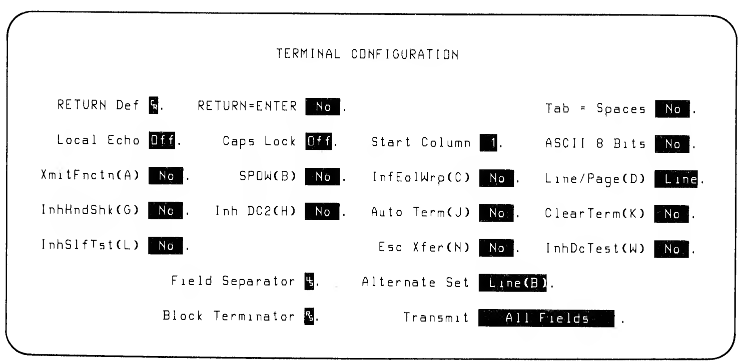
Table 6-4. Global Configuration Menu Fields (continued)

FIELD	FUNCTION
Graph Compat	Enables selection of graphics Compatibility mode, the Tektronix terminal type with which this terminal's operation is to be compatible, and scaled or unscaled operation. Off: Standard HP graphics operation. Unscaled: Compatibility mode, 4010 Unscaled mode. Scaled: Compatibility mode, 4010 Scaled mode. Uns 4014: Compatibility mode, 4014 Unscaled mode. Scl 4014: Compatibility mode, 4014 Scaled mode. Default=Off

Terminal Configuration

Figure 6-2 illustrates the terminal configuration menu and the default values.

Figure 6-2. Terminal Configuration Menu



Except when the cursor is positioned in the fields labeled "RETURN Def", "Field Separator", "Start Column", or "Block Terminator", the alphanumeric keys are

disabled and you select the desired parameters using the NEXT CHOICE and PREVIOUS CHOICE function keys.

The meanings of the various fields are described in table 6-5.

Table 6-5. Terminal Configuration Menu Fields

FIELD	FUNCTION
`RETURN Def`	This field specifies a one or two character sequence for the [Return] key, which will be sent to the host computer whenever the [Return] key is pressed. If the second character is a space, it is ignored. (If a control character is to be used in the sequence, it will be necessary to use the DISPLAY FUNCTNS key to enter it.) Values: Any one or two character sequence Default=`<CR>`
`RETURN=ENTER`	This field specifies whether or not you want the [Return] key to function as though it were the [Enter] key. The value "`Yes`" causes both keys to function in the manner currently defined for the [Enter] key when the terminal is in Remote mode. This can be useful when running VIEW/3000-based software or other block mode programs. `Yes`: [Return] key equals [Enter] key `No`: [Return] key not equal to [Enter] key Default=`No`
`Tab=Spaces`	When this feature is enabled, pressing the [Tab] key generates the number of ASCII space codes required to move the cursor forward to the next tab stop. If no tab stops exist between the current cursor position and the end of the line, the bell sounds and no spaces are generated. Similarly, pressing the [Tab] and [Shift] keys together generates the number of ASCII backspace codes required to move the cursor backward to the preceding tab stop (if the cursor is already located at the left margin when the backtab is attempted, the bell sounds and no backspaces are generated). Note that when operating in Local mode this function actually changes data characters within alphanumeric memory to spaces. In Remote mode, the spaces are transmitted over the datacomm port and the data

(continued)

Table 6-5. Terminal Configuration Menu Fields (continued)

FIELD	FUNCTION
Tab=Spaces (continued)	characters in the workspace are not changed unless the spaces are echoed back (either locally or from the host computer). When the feature is disabled and the terminal is in Remote mode, an ASCII "tab" character (decimal 9) is sent to the computer. In Local mode, the cursor is moved to the next tab stop. Yes: Tab=Spaces enabled No: Tab=Spaces disabled Default=No
Local Echo	This field specifies whether characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. On: Characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. Off: Characters entered through the keyboard are transmitted to the host computer only (if they are to appear on the screen, the host computer must "echo" them back to the terminal). Default=Off
Caps Lock	This field specifies whether the terminal generates the full 128-character ASCII set or only Teletype-compatible codes. On: The terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents. The {, |, and } keys generate the codes for [, \\, and], respectively. The key for generating ~ and \\ is disabled. Off: The terminal generates the full 128-character ASCII set of codes. Default=Off
Start Column	Under a very specific set of circumstances, when you enter data through the keyboard the terminal remembers, for each line, the position of the leftmost character you entered. This is accomplished using a logical start-of-text pointer, which is stored with the line in display memory.

(continued)

Table 6-5. Terminal Configuration Menu Fields (continued)

FIELD	FUNCTION
`Start Column` (continued)	The logical start-of-text pointer is generated only when both of the following conditions are true: **1.** The terminal is in any mode except Line Modify or Modify All (Remote, Local, Character, Block, or Format mode). **2.** The line in which you are entering data is the bottommost used line in display memory (there are no printing or non-printing characters following this line in display memory). When you are operating in Line Modify or Modify All mode and you press [Enter] or [Return], transmission of the line from the terminal normally begins at the logical start-of-text pointer in the line. If the line has no logical start-of-text pointer, transmission begins at the column selected in the "`Start Column`" field of the Terminal Configuration menu. (The identity of this column is saved in nonvolatile memory.) The active value of the "`Start Column`" field can also be temporarily redefined using one of the "margin/tab/col" function keys. Values: 1–160 (must be less than or equal to the value selected in the "`Columns`" field of the Global Configuration menu) Default=1
`ASCII 8 Bits`	When this operating mode is enabled (`Yes`), the terminal transmits 8-bit ASCII codes in which the eighth (high-order) bit, when set (1), indicates that the character is from the Roman Extension character set. This is a Hewlett-Packard convention, ordinarily used only when communicating with an HP 300 computer system, an HP 3000 (MPE 5) computer system, or in conjunction with certain HP line printers (such as the HP 2635A Printing Terminal). If set to `Yes`, the "`Parity/DataBits`" field on the Remote Datacomm menu must be set to "`None/8`". If set to `No`, the Remote Datacomm menu values aren't significant to the effects of this field. **NOTE** If the Remote Datacomm menu is not set up as indicated above, stripping of the 8th bit will occur, resulting in some characters being changed during data transfer. Also, when the terminal is set for 8-bit transfer, the host computer must also be set up for 8-bit data transfer.

(continued)

Table 6-5. Terminal Configuration Menu Fields (continued)

FIELD	FUNCTION
ASCII 8 Bits (continued)	Yes: 8-bit codes. No: Standard 7-bit codes. Default=No
XmitFnctn(A)	This field specifies whether escape code functions are both executed at the terminal and transmitted to the host computer. Yes: The escape code sequences generated by control keys such as Next and Prev are transmitted to the host computer. If local echo is On, the function is also performed locally. No: The escape code sequences for the major function keys are executed locally but not transmitted to the host computer. Note that turning on display functions generates an "Ec Y" to the host computer; turning it off generates an "Ec Z". Default=No
SPOW(B)	This field specifies whether or not spaces entered through the keyboard will overwrite existing characters. No: Spaces entered through the keyboard will overwrite existing characters. Yes: Enable Space OverWrite (SPOW) latch. When the SPOW latch is off, overwriting occurs. When the SPOW latch is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters. The SPOW latch is turned on by a carriage return and is turned off by a line feed, home up, tab, or back tab. Default=No
InhEolWrp(C)	Specifies whether or not end-of-line wrap is inhibited. No: When the cursor reaches the right margin it automatically moves to the left margin in the next lower line (a local carriage return and line feed are generated).

(continued)

Table 6-5. Terminal Configuration Menu Fields (continued)

FIELD	FUNCTION
InhEolWrp(C) (continued)	Yes: When the cursor reaches the right margin it remains in that screen column until an explicit carriage return or other cursor movement function is performed (succeeding characters overwrite the existing character in that screen column). Default=No
Line/Page(D)	This field specifies whether or not the terminal, when operating in Block mode, will transmit data a line at a time or a page at a time. Line: When operating in Block mode, the terminal will transmit data a line at a time. Page: When operating in Block mode, the terminal will transmit data a page at a time. For a detailed description of the differences between Block Line and Block Page mode, refer to "ENTER Key Data Transfers" in Section 10 of this manual. Default=Line
InhHndShk(G) and Inh DC2(H)	Together, these fields determine what type of handshaking is to be used when transferring blocks of data from the terminal to the host computer. There are four types of block data transfers: ■ Initiated by the [Enter] key. ■ Send display (Ec d). ■ Definition string transfer for a function key ("T" type) when the function key is pressed. ■ Status transfer resulting from a status request. Refer to Section 10 for detailed information on handshaking. Default=No (both fields)
AutoTerm(J)	This parameter is effective only when the [Enter] key is pressed, in Block mode. Selects the start and stop points of the block of data to be transferred. Refer to Section 10 for more detail.

(continued)

Table 6-5. Terminal Configuration Menu Fields (continued)

FIELD	FUNCTION
AutoTerm(J) (continued)	Yes: Insert a non-displaying terminator (NDT) at the current cursor position, then move the cursor back to the previous non-displaying terminator. (If none is found, the cursor is moved to the "home" position.) Data between the resulting cursor position and the entered NDT is transferred. No: A non-displaying terminator is not inserted and the cursor is not moved. Data between the cursor position and the next block terminator or NDT is transferred. Default=No
ClearTerm(K)	Clears, or doesn't clear, a non-displaying terminator, if the display transfer operation is ended by encountering a non-displaying terminator. Yes: clear terminator. No: do not clear terminator. Default=No
InhSlfTst(L)	Enables or disables terminal self test. Pressing the TERMINAL TEST or POWER ON TEST function key or issuing an "Ec z" results in an error message. Other tests are not affected by this field. Yes: test disabled. No: test enabled. Default=No
Esc Xfer(N)	Enables or disables transfer, to the external printer, of escape codes embedded in data. These escape codes might be those specifying display enhancements, Format mode fields, and alternate character sets. Yes: Escape code transfer enabled. No: Escape code transfer disabled. Default=No
InhDcTst(W)	Enables or disables datacomm self test. Pressing the DATACOMM TEST function key, or initiating an "Ec x", when the test is disabled, results in an error message.

(continued)

Table 6-5. Terminal Configuration Menu Fields (continued)

FIELD	FUNCTION
InhDcTst(W) (continued)	Yes: datacomm self test disabled. No: datacomm self test enabled. Default=No
Field Separator	When the [Enter] key is pressed while the terminal is displaying a formatted display in Block Page mode, the terminal automatically transmits the specified field separator character at the end of each unprotected field (except the final one). Value: Any ASCII character. Default: <US>.
Alternate Set	Selects the current alternate character set. The alternate character set is selected when the terminal receives a Shift Out (<SO>). Then, to return to the base character set, the terminal must receive a Shift In (<SI>). Base(@): Base set (ASCII). Math(A): Math set. Line(B): Line-drawing character set. Italic(E): Italic ASCII set. Bold(F): Bold ASCII set. Default=Line(B)
Block Terminator	For data transfers between the terminal and a host computer, the terminal (under certain circumstances) transmits the specified block terminator character at the end of the transfer operation. For details, see "ENTER Key Data Transfers" in Section 10. This character, when encountered in display memory, terminates a data transfer ("copy" device control operations and [Enter] key transmissions). Value: Any single ASCII character. Default: <RS>.

(continued)

Table 6-5. Terminal Configuration Menu Fields (continued)

FIELD	FUNCTION
`Transmit`	Specifies whether, in Format mode, to transmit all fields, or only those which have been modified. `All Fields:` Transmit all fields. `Modified Fields:` Transmit only modified fields. Default=`All Fields`

Datacomm Configuration

The two datacomm menus, the Remote Full Duplex Hardwired and the Remote Full Duplex Modem Configuration menus, are used to configure the port (port 1 or 2) selected as the remote port in the "`Remote/Serial Dev`" field of the Global Configuration menu. If the host computer is an HP 1000 or HP 3000, the port can be configured for the commonly-used values by pressing the HP 1000 PT. PT. or HP 3000 PT. PT. key.

Figures 6-3 and 6-4 illustrate the menus, and table 6-6 describes the menu fields. The fields are the same for both menus, except that the modem menu lacks a "`CS(CB)XMIT`" field. The default values for for both menus are shown in the figures. Refer to Section 8 for further information on data communications.

Figure 6-3. Hardwired Datacomm Menu

```
            REMOTE DATACOMM Full Duplex Hardwired CONFIGURATION          Port 1

BaudRate  9600 .  Parity/DataBits None/8.   Check Parity  No .

Asterisk  Off .         Stop Bits 1.              EnqAck  Yes.

 TR(CD) Hi.               SR(CH) Lo.

RecvPace  None  .   Break Time 160.         RR(CF)Recv  No .

XmitPace  None  . Stop Function Xon/Xoff.   CS(CB)Xmit  No .  DM(CC)Xmit  No .
```

Figure 6-4. Modem Datacomm Menu

```
            REMOTE DATACOMM Full Duplex Modem CONFIGURATION              Port 1

BaudRate  1200.  Parity/DataBits 0's/7 .   Check Parity  No .

Asterisk   RR .         Stop Bits 1.              EnqAck  Yes.

 TR(CD) Hi.               SR(CH) Lo.

RecvPace  None  .   Break Time 160.         RR(CF)Recv  No .

XmitPace  None  . Stop Function Xon/Xoff.                    DM(CC)Xmit  No .
```

Table 6-6. Remote Datacomm Configuration Menu Fields

FIELD	FUNCTION
`Port`	Indicates the port (1 or 2) selected as the remote datacomm port, to which the values on this menu apply. This field is for information only; no entry can be made in it.
`BaudRate`	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: `110` `134.5` `150` `300` `600` `1200` `1800` `2400` `4800` `9600` `19.2K` Default=`9600` (Hardwired) `1200` (Modem)
`Parity/DataBits`	This field specifies the type of parity and number of valid-data bits in each byte. `None/8`: No parity bit, 8 data bits `0's/7`: Parity bit (eighth bit) always zero `Odd/7`: Odd parity `1's/7`: Parity bit always one `Even/7`: Even parity Defaults: `None/8` (hardwired menu) `0's/7` (modem menu)
`Check Parity`	Together with the "`Parity/DataBits`" field, selects checking or ignoring of parity for each received data byte. If the "`Parity/DataBits`" field is set to "`None/8`", "`0's/7`", or "`1's/7`" and this field is set to "`Yes`", an "`Invalid Configuration`" error message is displayed. `Yes`: Parity checked `No`: Parity not checked Default=`No` (both menus)

(continued)

Table 6-6. Remote Datacomm Configuration Menu Fields (continued)

FIELD	FUNCTION
Asterisk	Specifies whether the transmit indicator should be enabled or disabled and, if enabled, which datacomm control line it should represent. An asterisk reflects an active state of the line, and no asterisk, the inactive state. Off: No indicator DM: Data Mode (DM) or RS-232C Data Set Ready (CC) RR: Receiver Ready (RR) or RS-232C Carrier Detect (CF) CS: Clear to Send (CS) or RS-232C Clear to Send (CB) Default= Off (hardwired) RR (modem)
Stop Bits	This field specifies the number of "stop bits" to be appended to each data character transmitted by the terminal. (Received data is accepted with one or two stop bits regardless of the setting of this field.) 1: One stop bit appended. 2: Two stop bits appended. Default=1 (both menus)
EnqAck	This field enables or disables the use of the Hewlett-Packard ENQ-ACK handshake (described under "Pacing Mechanisms" in Section 8). Yes: Enabled No: Disabled Default=Yes (both menus)
TR(CD)	This field specifies the desired state of the RS-232C TR (Data Terminal Ready) line when the terminal is first turned on or reset. The TR signal line is used to control receive pacing. When the terminal performs a disconnect, it returns the TR line to the state specified by this field. Values: Hi Lo Default=Hi (both menus)

(continued)

Table 6-6. Remote Datacomm Configuration Menu Fields (continued)

FIELD	FUNCTION
SR(CH)	Specifies the desired state of the RS-232C Data Signal Rate Selector (CH) line when the terminal is powered up or when the terminal is reset. This line is normally used on dual speed modems to select the appropriate speed (single speed modems ignore this line). Values: Hi Lo Default=Lo (both menus)
RecvPace	Selects the type of receive pacing to be used. Receive pacing is a mechanism by which the terminal automatically controls (halts and resumes) the transmission of data from the remote device. Receive pacing selected through this field is different from, and independent of, ENQ/ACK pacing. If this field is set to "Xon/Xoff", the terminal will automatically perform receive pacing using Xon (ASCII <DC1>) and Xoff (ASCII <DC3>) control codes. With this type of receive pacing, the terminal causes the remote device to halt transmission by sending an Xoff code. It resumes transmission by sending an Xon code. For this type of receive pacing to work, the remote device must, of course, be configured to start and stop transmission in response to Xon and Xoff codes. If the remote device recognizes Xon and Xoff codes and the terminal is operating in Character mode, you can issue the codes through the keyboard regardless of the setting of this field. The CTRL and Q (or q) keys (when pressed simultaneously) generate an Xon code and the CTRL and S (or s) keys generate an Xoff. If this field is set to "TR(CD)", data receipt is controlled by the Data Terminal Ready signal line (refer to the "TR(CD)" field discussion). When, while receiving, the terminal reaches a state in which it can no longer receive, it deactivates the TR line to stop transmission. Then, when it is ready to receive, it reactivates the TR signal line.

(continued)

Table 6-6. Remote Datacomm Configuration Menu Fields (continued)

FIELD	FUNCTION
	`None:` No receive pacing. `Xon/Xoff:` `Xoff` stops transmission; `Xon` resumes transmission. `TR(CD):` Deactivates TR line to stop transmission, and activates it to resume transmission. Default=`None` (both menus)
`Break Time`	Selects the duration, in milliseconds, of the "Break" character pulse when the Break key is pressed. The HP 3000 computer requires a duration of approximately 160 ms (the default value). However, for some systems, a duration of this length not only breaks contact with the computer but with the entire network in which the terminal is connected. For such systems, contact with the network can be maintained, while breaking contact with the computer, by using a shorter duration. The duration to use depends on the system. Values: 100 to 250 Default=`160` (both menus)
`RR(CF)Recv`	This field specifies whether or not a high (active) state on the RS-232C Received Line Signal Detector (CF) control line is required to receive data. `Yes:` CF line must be high `No:` CF line not used Default=`No` (both menus)
`XmitPace`	Selects whether transmit pacing is to be used. Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using `Xon` and `Xoff` control codes. When the terminal receives an `Xoff` (ASCII `<DC3>`), it stops transmitting data. When the terminal subsequently receives an Xon code (ASCII `<DC1>`), it resumes transmitting data.

(continued)

Table 6-6. Remote Datacomm Configuration Menu Fields (continued)

FIELD	FUNCTION
	NOTE When transmit pacing is enabled, it may interfere with the operation of DC1/DC2 pacing as specified by the "InhHndShk(G)" and "Inh DC2(H)" fields on the Terminal Configuration menu. If Xon/Xoff transmit pacing is enabled, DC1 is treated as an Xon only when a DC3 was received previously. Also, since the DC1 character is used for both Xon and the handshake trigger, the host program must take care to distinguish between the two. If this field is set to "None", the terminal does not recognize the ASCII <DC1> and <DC3> codes as Xon and Xoff. For other forms of transmit pacing, refer to the description of the "CS(CB)Xmit", and "DM(CC) Xmit" fields. None: No transmit pacing used. Xon/Xoff: Terminal stops its transmission when it receives an Xoff. Resumes transmission when it receives an Xon. Default=None (both menus)
Stop Function	Enables user control of terminal transmit/receive pacing through use of the Stop key. **NOTE** When either "Xon/Xoff" or "Hold Ack" is selected on the menu, and the terminal has been set to the Stop condition by the Stop key, all input is ignored until the Stop key is pressed again. The terminal is in a state of soft reset, hard reset, and break. Xon/Xoff: Sends, on alternate keystrokes of the Stop key, Xoff and Xon to the host computer. Xoff is sent only after the terminal input buffer fills up, not when the key is struck. Hold Ack: When the Stop key is pressed, the "Ack" response to the next-received "Enq" is held until the Stop key is pressed again.

(continued)

Table 6-6. Remote Datacomm Configuration Menu Fields (continued)

FIELD	FUNCTION
	None: Stop key has no effect. Defaults (both menus): Power on— Xon/Xoff HP 3000 PT. PT. key— Xon/Xoff HP 1000 PT. PT. key— Hold Ack
CS(CB)Xmit	Specifies whether or not a high (active) state on the RS-232C Clear to Send (CB) control line is required for the terminal to transmit data. Yes: Clear to Send line must be high. No: Clear to Send signal ignored. Default=No
DM(CC)Xmit	This field specifies whether a high (active) state on the RS-232C Data Set Ready (CC) control line is required for the terminal to transmit data. Yes: Data Set Ready signal must be high for transmission. No: Data Set Ready signal ignored. Default=No (both menus)

External Serial Device Configuration

The External Serial Device Configuration menu is used to configure the port selected as the external port in the "Remote/External" field of the Global Configuration menu. Figure 6-5 illustrates the menu, with the default values selected. The menu fields are described in table 6-7.

Figure 6-5. External Serial Device Configuration Menu

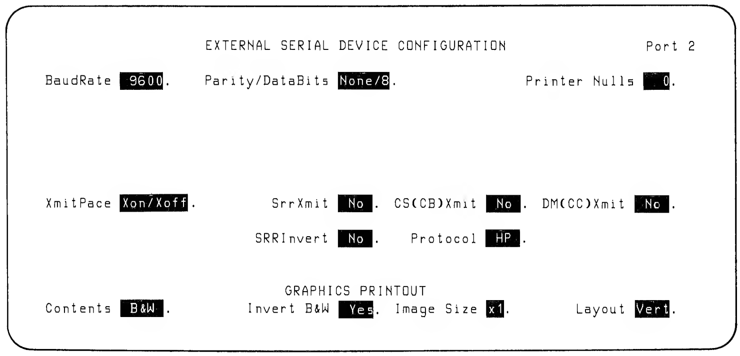


Table 6-7. External Serial Device Configuration Menu Fields

FIELD	FUNCTION
Port	Indicates the port (1 or 2) selected as the external port, to which the values on this menu apply. This field is for information only; no entry should be made in it by the user.
BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19.2K Default=9600

Table 6-7. External Serial Device Configuration Menu Fields (continued)

FIELD	FUNCTION
`Parity/DataBits`	This field specifies the type of parity and number of valid data bits in each byte. (Bit 8 is the parity bit.) `None/8:` No parity bit, 8 data bits `0's/7:` Parity bit (eighth bit) always zero `Odd/7:` Odd parity `1's/7:` Parity bit always one `Even/7:` Even parity `Default: None/8`
`Printer Nulls`	Specifies the number of ASCII null codes to be transmitted to a hardcopy device after each ASCII control code. Values: 0 to 255 Default=0
`XmitPace`	Selects whether transmit pacing is to be used. Transmit pacing is a mechanism by which the external serial device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using `Xon` and `Xoff` control codes. When the terminal receives an `Xoff` (ASCII `<DC3>`), it stops transmitting data. When the terminal subsequently receives an `Xon` code (ASCII `<DC1>`), it resumes transmitting data. If this field is set to "`None`", the terminal does not recognize the ASCII `<DC1>` and `<DC3>` codes as `Xon` and `Xoff`. `None:` No transmit pacing used. `Xon/Xoff:` Terminal stops its transmission when it receives an `Xoff`. Resumes transmission when it receives an `Xon`. Default=`Xon/Xoff`

(continued)

Table 6-6. Remote Datacomm Configuration Menu Fields (continued)

FIELD	FUNCTION
SRRXmit	This field specifies whether an active (high) state on the RS-232C Secondary Receiver Ready (SCF) control line is required for transmitting data. This mechanism is primarily used with hardcopy device which must be able to control the transmission of data from other devices. The control line is connected to RS-232C, pin 12. Yes: Secondary Receiver Ready line must be high for transmission. No: Secondary Receiver Ready signal ignored. Default=No
CS(CB)Xmit	Specifies whether or not an active (high) state on the RS-232C Clear to Send (CB) control line is required for the terminal to transmit data. Yes: Clear to Send signal must be active. No: Clear to Send signal ignored. Default=No
DM(CC)Xmit	This field specifies whether an active (high) state on the RS-232C Data Set Ready (CC) control line is required for the terminal to transmit data. Yes: Data Set Ready signal must be active for transmission. No: Data Set Ready signal ignored. Default=No
SRRInvert	This field applies only when the "SRRXmit" field is set to "Yes". When both the "SRRXmit" and "SRRInvert" fields are set to "Yes", the active state of the RS-232C Secondary Receiver Ready (SCF) control line is detected as the most negative voltage level instead of the most positive voltage level. Yes: Active level of the Secondary Receiver Ready signal selected as most negative voltage. No: Active level of signal remains most positive voltage. Default=No

Table 6-6. Remote Datacomm Configuration Menu Fields (continued)

FIELD	FUNCTION
Protocol	Selects the printer protocol: HP or Diablo. The protocol determines the set of initialization commands the terminal sends to the printer when certain function keys (PRESET PRINTER, EXPAND PRINT, COMPRESS PRINT, REPORT PRINT, and METRIC PRINT) are used or when a raster dump of graphics memory is requested. Refer to Section 7, External Devices, for details. `HP:` HP protocol. `Other:` Diablo C150 color printer protocol. Default: `HP`
Contents	Selects either black and white or color graphics data to be sent to the hardcopy device. (This field applies only to color terminals.) `B&W:` the bits from all raster planes are ORed together, then transmitted. `Color:` data is transmitted in color format. Default=`B&W`
Invert B&W	Selects whether black and white are to be printed as they appear on the screen or inverted, with screen black printed as black and screen white printed as white or vice-versa.

MENU SELECTION	SCREEN BACKGROUND	COPY BACKGROUND
Yes	Black	Paper color
Yes	White	Black
No	Black	Black
No	White	Paper color

Values: `Yes`
`No`

Default=`Yes`

(continued)

Table 6-7. External Serial Device Configuration Menu Fields (continued)

FIELD	FUNCTION
`ImageSize`	Selects the size of the graphics image printed by the hardcopy device. `x1:` each raster memory pixel produces one printed pixel. `x2:` each raster memory pixel produces two horizontal and two vertical printed pixels, for a total of four printed pixels for each raster pixel. This produces a printed picture twice the size (in both the vertical and horizontal dimensions) of the the picture produced for the "`x1`" selection. For printers which lack the width for double wide images, the image will be truncated in the horizontal dimension. If either Report or Metric mode is selected (Device Modes set of function keys) and there isn't enough room left on the current printer page to accommodate the screen image, the upper part of the image will be copied to the current page and the remainder will be copied to the following page, separated by several blank lines bordering the perforation. By turning off Report and Metric modes, the image can be copied across the perforation without being split by blank lines. But to copy the whole image onto a single sheet of paper, the paper must be advanced to the start of a fresh sheet before copying the screen image. Default=`x1`
`Layout`	Selects whether raster dumps to the hardcopy device are to be copied vertically, as viewed on the screen, or horizontally. `Vert:` Image is copied vertically; raster rows are sent to the hardcopy device. The copied picture has the same orientation as the screen picture. `Hor:` Image is copied horizontally; raster columns are sent to the printer. Default=`Vert`

External Parallel Device Configuration

The External Parallel Device Configuration menu (figure 6-6) is used for configuring Centronics-compatible hardcopy devices, which receive data in parallel rather than serial form. The menu fields are described in table 6-8.

Figure 6-6. External Parallel Device Configuration Menu

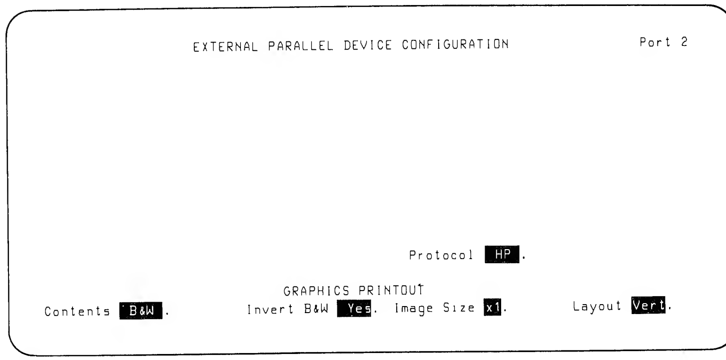


Table 6-8. External Parallel Device Configuration Menu Fields

FIELD	FUNCTION
Port	Indicates port 2 is selected as the external port, to which the values on this menu apply. This field is for information only; no entry should be made in it by the user.
Protocol	Selects the printer protocol: HP or Diablo. The protocol determines the set of initialization commands the terminal sends to the printer when certain function keys (PRESET PRINTER, EXPAND PRINT, COMPRESS PRINT, REPORT PRINT, and METRIC PRINT) are used or when a raster dump of graphics memory is requested. Refer to Section 7, External Devices, for details. HP: HP protocol. Other: Diablo C150 color printer protocol. Default: HP

(continued)

Table 6-8. External Parallel Device Configuration Menu Fields (continued)

FIELD	FUNCTION
Contents	Selects either black and white or color graphics data to be sent to the hardcopy device. (This field applies only to color terminals.) B&W: the bits from all raster planes are ORed together, then transmitted. Color: data is transmitted in color format. Default=B&W
Invert B&W	Selects whether black and white are to be printed as they appear on the screen or inverted, with screen black printed as black and screen white printed as white or vice-versa. MENU SELECTION / SCREEN BACKGROUND / COPY BACKGROUND Yes / Black / Paper color Yes / White / Black No / Black / Black No / White / Paper color Values: Yes No Default=Yes
ImageSize	Selects the size of the graphics image printed by the hardcopy device. x1: each raster memory pixel produces one printed pixel. x2: each raster memory pixel produces two horizontal and two vertical printed pixels, for a total of four printed pixels for each raster pixel. This produces a printed picture twice the size (in both the vertical and horizontal dimensions) of the the picture produced for the "x1" selection. For printers which lack the width for double wide images, the image will be truncated in the horizontal dimension. If either Report or Metric mode is selected (Device Modes set of function keys) and there isn't enough room left on the current

Table 6-8. External Parallel Device Configuration Menu Fields (continued)

FIELD	FUNCTION
`ImageSize` (continued)	printer page to accommodate the screen image, the upper part of the image will be copied to the current page and the remainder will be copied to the following page, separated by several blank lines bordering the perforation. By turning off Report and Metric modes, the image can be copied across the perforation without being split by blank lines. But to copy the whole image onto a single sheet of paper, the paper must be advanced to the start of a fresh sheet before copying the screen image. Default=`x1`
`Layout`	Selects whether raster dumps to the hardcopy device are to be copied vertically, as viewed on the screen, or horizontally. `Vert:` Image is copied vertically; raster rows are sent to the hardcopy device. The copied picture has the same orientation as the screen picture. `Hor:` Image is copied horizontally; raster columns are sent to the printer. Default=`Vert`

HP-IB Configuration

The External HP-IB Configuration menu (figure 6-7) can be displayed only if the terminal contains the HP-IB Auxilliary Port option. It is used for configuring the interface between the terminal and HP-IB-connected hardcopy devices. The menu fields are described in table 6-9.

Figure 6-7. External HP-IB Configuration Menu

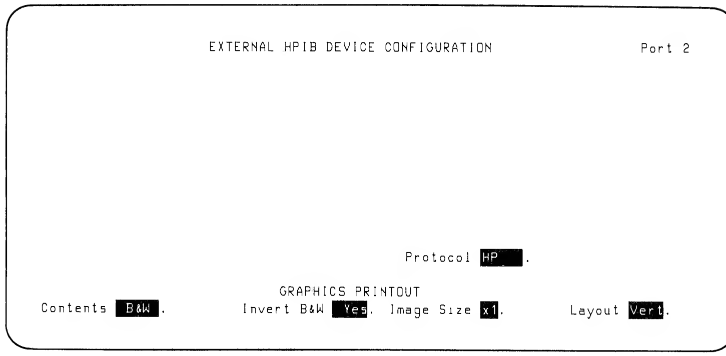


Table 6-9. External HP-IB Configuration Menu Fields

FIELD	FUNCTION
Port	Indicates port 2 is selected as the external port, to which the values on this menu apply. This field is for information only; no entry should be made in it by the user.
Protocol	Selects the printer protocol: HP or Diablo. The protocol determines the set of initialization commands the terminal sends to the printer when certain function keys (PRESET PRINTER, EXPAND PRINT, COMPRESS PRINT, REPORT PRINT, and METRIC PRINT) are used or when a raster dump of graphics memory is requested. Refer to Section 7, External Devices, for details. HP: HP protocol. Other: Diablo C150 color printer protocol. Default: HP

Table 6-9. External HP-IB Configuration Menu Fields (continued)

FIELD	FUNCTION
Contents	Selects either black and white or color graphics data to be sent to the hardcopy device. (This field applies only to color terminals.) B&W: the bits from all raster planes are ORed together, then transmitted. Color: data is transmitted in color format. Default=B&W
Invert B&W	Selects whether black and white are to be printed as they appear on the screen or inverted, with screen black printed as black and screen white printed as white or vice-versa. MENU SELECTION / SCREEN BACKGROUND / COPY BACKGROUND: Yes / Black / Paper color Yes / White / Black No / Black / Black No / White / Paper color Values: Yes No Default=Yes
ImageSize	Selects the size of the graphics image printed by the hardcopy device. x1: each raster memory pixel produces one printed pixel. x2: each raster memory pixel produces two horizontal and two vertical printed pixels, for a total of four printed pixels for each raster pixel. This produces a printed picture twice the size (in both the vertical and horizontal dimensions) of the the picture produced for the "x1" selection. For printers which lack the width for double wide images, the image will be truncated in the horizontal dimension. If either Report or Metric mode is selected (Device Modes set of function keys) and there isn't enough room left on the current printer page to accommodate the screen image, the upper part of

(continued)

Table 6-9. External HP-IB Configuration Menu Fields (continued)

FIELD	FUNCTION
`ImageSize` (continued)	the image will be copied to the current page and the remainder will be copied to the following page, separated by several blank lines bordering the perforation. By turning off Report and Metric modes, the image can be copied across the perforation without being split by blank lines. But to copy the whole image onto a single sheet of paper, the paper must be advanced to the start of a fresh sheet before copying the screen image. Default=`x1`
`Layout`	Selects whether raster dumps to the hardcopy device are to be copied vertically, as viewed on the screen, or horizontally. `Vert:` Image is copied vertically; raster rows are sent to the hardcopy device. `Hor:` Image is copied horizontally; raster columns are sent to the hardcopy device. Default=`Vert`

ANSI Configuration

The ANSI Configuration menu (figure 6-8) enables selection of parameters for use in ANSI and EM52 modes. Table 6-10 contains descriptions of each field.

Figure 6-8. ANSI Configuration Menu

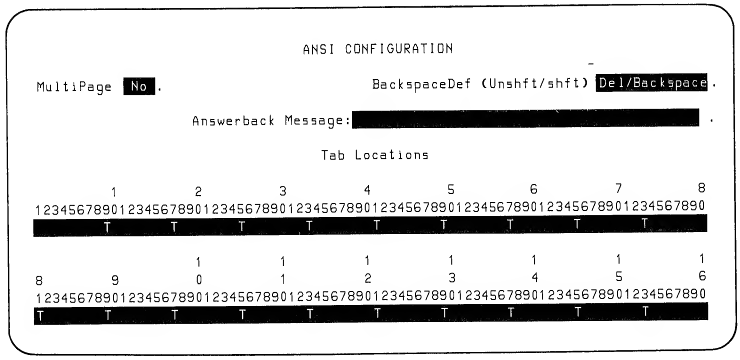


Table 6-10. ANSI Configuration Menu Fields

FIELD	FUNCTION
Multipage	Selects display memory size. Should be set to No for use with application programs designed for ANSI terminals with a single page of display memory. When set to No, the [Next], [Prev], [Shift] [↑] (roll up), and [Shift] [↓] (roll down) key operations are disabled. No: One page (24 lines). Yes: All available memory. Default=No
BackspaceDef (Unshft/shft)	Selects the operations for shifted and unshifted use of the [Back space] key. Backspace/Del: Unshifted = backspace Shifted = delete Del/Backspace: Unshifted = delete Shifted = backspace Default=Del/Backspace
Answerback Message	Displays the current answerback message and enables entry of a new one (up to 40 characters). The answerback message is transmitted automatically to the host computer when the terminal receives an "Enq" (hexadecimal 05) character from the computer.

(continued)

Table 6-10. ANSI Configuration Menu Fields (continued)

FIELD	FUNCTION
`Answerback Message` (continued)	If the answerback message field is not displayed, press the ANSWER BACK function key to display the field. To enter a new message, display the "`Answerback Message`" field and use the keyboard to enter the new message. The message can include control characters, which must be entered using the DISPLAY FUNCTNS key. Entered characters, including trailing blanks, are displayed in half-bright inverse video. Only the entered characters are transmitted. Values: Up to 40 alphanumeric or control characters. Default=No message supplied.
`Tab Locations`	Available for selecting tab stops. To set or reset tabs, position the cursor in the desired column and use either the NEXT CHOICE or PREVIOUS CHOICE key to toggle the tab on or off. The CLR ALL TABS key clears all tabs. Default=Tabs set every eight columns.

Programmatic Configuration

Escape sequences can be used to change the active values of some configuration menu parameters. Appendix A contains a list of the escape sequences, along with their functions. In addition to changing configuration values, the configuration menus can be locked and unlocked, using escape sequences.

Normally, escape sequences are transmitted to the terminal from a program operating on the host computer; however, they can also be generated at the keyboard.

Configuration Escape Codes

To set configuration parameters using escape codes, you must use an `Ec &k`, `Ec &s`, or `Ec)` sequence, depending on the parameters you wish to set.

A change of a parameter value using the `Ec &k` and `Ec &s` sequences takes effect immediately, but the content of nonvolatile memory is not changed. If a configuration menu is displayed on the screen when the escape sequence is received, the sequence is not executed until the menu is exited.

Lock/Unlock Configuration Menus

Using an escape sequence, you can "lock" the current configuration menus so that the menu can not be altered from the keyboard. Any attempt to access a locked menu from the keyboard will result in a "beep" from the bell and the message "`Function locked`". Note that when the configuration menus are locked, the **MODIFY ALL**, **BLOCK MODE**, **REMOTE MODE**, and **AUTO LF** mode selection keys are also locked.

To lock the menus, use the following escape sequence:

```
Ec &q 1L
```

To unlock the menus, use the following escape sequence:

```
Ec &q 0L
```

7 External Devices

Introduction

This section covers the following topics:

- Interface modules —A description of the interface modules, available as options, for connecting external devices to port 2.
- Device definitions —A description of devices.
- Device connections —Lists plotters and printers useable with the terminal as an "external" device. Supplies installation information for plotters, printers, and HP-IB networks.
- Device selection
- Printer control —Printer modes and control.
- HP-IB network control —HP-IB device control.
- Data transfer —Transfer of data between the terminal, computer, and external devices.
- Determining if a command was completed. —Determining if an escape sequence was successfully completed.

Interface Modules

Three mutually-exclusive terminal options, 092, 093, and 046, are available for connecting plotters and printers to the terminal. Each option supplies an interface module, through which devices are connected to port 2. The use of each option is as follows:

OPTION	USE
092	RS-232C connection. Can be used for a printer, plotter, or can be connected to the host computer, in place of port 1. For terminals with option 092, either port 1 or port 2 can be connected to the host computer, with the other port connected to a plotter or printer. The choice of port to connect to the computer, as opposed to the external device, is made in the "Remote/Serial Dev" field of the Global Configuration menu.
093	8-bit parallel Centronics-type printer connection.
046	HP-IB connection, for connection to an HP-IB network.

Device Definitions

A "device" is logically a data source, destination, or both. Most devices are identified by a device code. This is the number used to identify the device in data transfer escape sequences. Three of the devices (datacomm port, alphanumeric memory, and graphics memory) are built into the terminal. The "external device", a printer or plotter, must be connected to the terminal through an optional interface module.

The following paragraphs describe the device types. The device code is shown in parenthesis following the paragraph title.

Datacomm Port

A datacomm port, connected to a host computer, can be considered as a special device to which no device code has been assigned. It is the default source (or destination) device when no other device has been selected as the source (or destination). Normally, port 1 is used as the datacomm port. An exception exists when an RS232 interface module is installed in port 2 and the "`Remote/Serial Dev`" field on the Global Configuration menu is set to "`Port2/Port1`". In this case, port 2 is the datacomm port and port 1 is the external device port.

Alphanumeric Display Memory (3)

This is 1 to 12 pages of standard alphanumeric display memory.

Graphics Display Memory (7)

This is the 512x390 or 640x400 graphics raster memory.

External Device (4, 5, or 6)

For this terminal, the term "external device" refers to the printer or plotter connected to port 2 through an interface module. The interface can be one of the following:

- RS232C serial interface.
- Centronics-type parallel interface.
- HP-IB parallel interface.

Devices connected through an HP-IB parallel interface are a special class of external device. Refer to "HP-IB Network Device (5)", later in this section.

When no interface module is installed in port 2, the terminal ignores all escape sequences relating to an external device.

If an RS232C serial interface module is installed in port 2 and the "Remote/Serial Dev" field in the Global Configuration menu is set to "Port2/Port1", the terminal sees the device connected to port 1 as the "external device", and all escape sequences directed at the external device affect the port 1 device.

Generally, device codes 4, 5, and 6 can be used interchangeably for escape code parameters "s","d" or "u", regardless of the interface module installed in port 2.

An exception exists when an escape sequence which can only apply to an HP-IB network, such as changing the address of an HP-IB network device, is sent. In this case, device code 5 must be used and an HP-IB device must be installed for successful execution of the sequence.

HP-IB Network Device (5)

This is a special class of external device. When a device, such as a plotter, is connected to the HP-IB network, host computer software can command it to return certain data, such as status. Both a printer and a plotter can be connected as part of the HP-IB network, each at a different HP-IB address.

Data Transfer Possibilities

Data transfer and control operations take place only in certain directions and between certain devices. Table 7-1 shows the possibilities.

Table 7-1. Data Transfer Possibilities

LEGEND:
- YES —Yes, data transfer possible.
- NO —No, data transfer not possible.
- N/A —Not applicable, case invalid.
- (x) —"x" is the number used in the data transfer escape sequence to select the associated device.

SOURCE DEVICE	DESTINATION DEVICE			
	Alphanumeric Display (3)	External Device (4/5/6)	HP-IB Device (5)	Computer Datacomm Port
Alphanumeric Display (3)	NO	YES	YES	YES
HP-IB Network Device (5)	NO	N/A	N/A	YES
Graphics Display (7)	NO	YES	YES	NO
Computer Datacomm Port	YES	YES	YES	NO

Supported Printers

Table 7-2 lists the printers supported by the terminal.

Table 7-2. Supported Printers

Notes: 1. Numbers in parenthesis are printer option numbers.

2. For all listed printers except the Diablo C150, the "`Protocol`" field on the External Device Configuration menu must be set to "`HP`". For the Diablo C150, the field must be set to "`Other`".

PRINTER TYPE PORT 2 OPTION	ALPHANUMERIC PRINTERS	B&W GRAPHICS PRINTERS	COLOR GRAPHICS PRINTERS
RS232C Option 092	HP 2601A HP 2602A HP 2631B HP 2671A (040) HP 82905B (240) (340) (440)	HP ThinkJet: 2225D HP 2671G (040) HP 2673A (040) HP 2932A HP 2934A HP 2686A	
HP-IB Option 046	HP 2602A (046) HP 2631B (046) HP 2671A HP 82905B (002) (003) (004)	HP 2631G HP 2671G HP 2673A HP 2932A (046) HP 2934A (046) HP 82906A HP ThinkJet: 2225A	
Centronics Option 093	HP 2671A(042)	HP ThinkJet: 2225C HP 2671G (042) HP 2932A (042) HP 2934A (042)	Diablo C150

Supported Plotters

RS232C plotters, supported in an eavesdrop connection to the datacomm port, or in pass-through connection as an external device are listed below:

HP 7220C/T
HP 7221C/T
HP 7470A, options 001 and 016
HP 7475A, options 001 and 016
HP 7580B
HP 7585B

Following is a list of supported HP-IB plotters. These plotters are connected, as an external device, in a passthrough connection, to port 2, using the option 046 (HP-IB) interface module.

HP 7470A, option 002
HP 7475A, option 002
HP 7580B
HP 7585B
HP 9872C/T

Installation

The following paragraphs list and illustrate the possibilities for connecting an external device to the terminal, describe the two types of plotter connection, and list the connecting cables.

External Device Connections

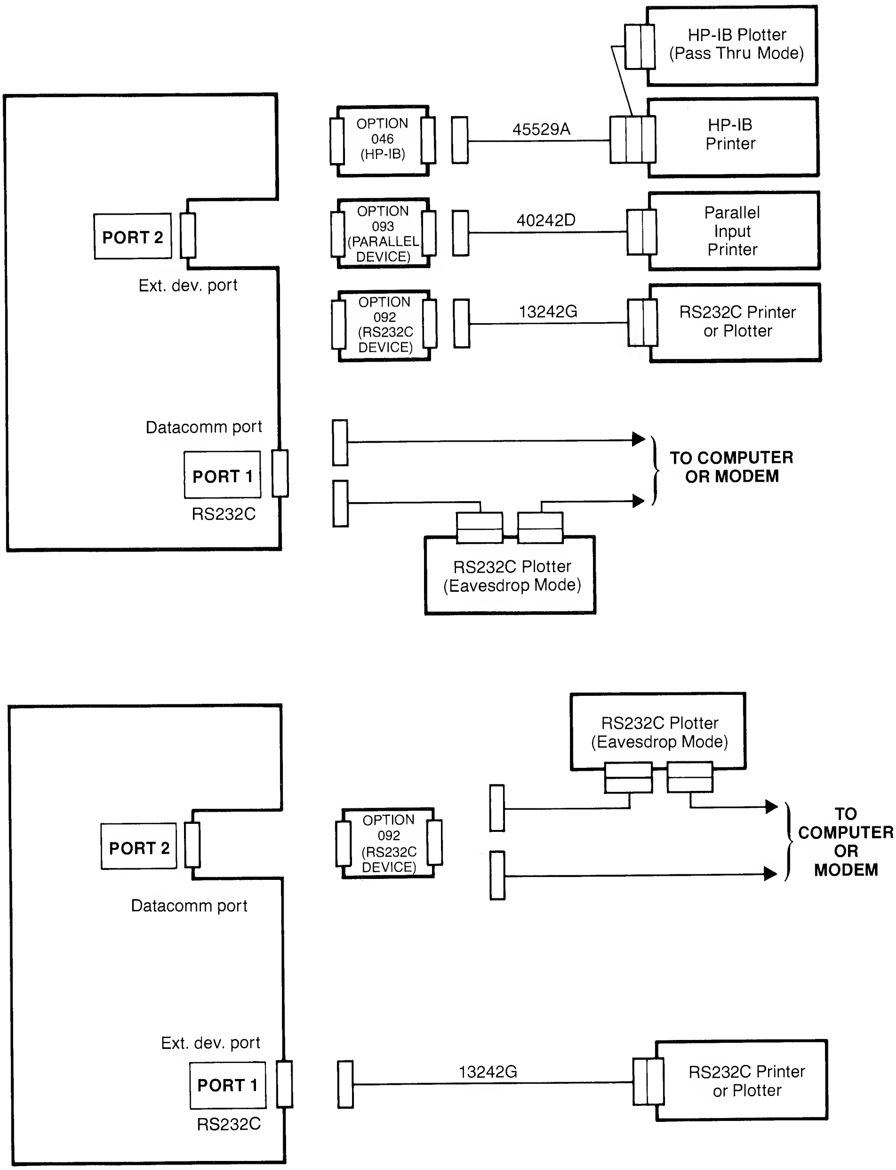
External device to terminal connections are listed in table 7-3.

Table 7-3. Terminal External Device Connection Capabilities

PORT 2 OPTION	PORT	GLOBAL CONFIG. MENU "Remote/Serial Dev" FIELD	CONNECTED TO
None	1	Field not displayed.	Computer, optionally with a plotter in eavesdrop mode.
	2	Field not displayed.	Not connected.
092	1	PORT1/PORT2	Computer, optionally with a plotter in eavesdrop mode.
	2	PORT1/PORT2	RS-232 printer or plotter.
092	1	PORT2/PORT1	RS-232 printer or plotter.
	2	PORT2/PORT1	Computer, optionally with a plotter in eavesdrop mode.
093	1	Field not displayed.	Computer, optionally with a plotter in eavesdrop mode.
	2	Field not displayed.	Centronics-type (parallel) printer.
046	1	Field not displayed.	Computer, optionally with a plotter in eavesdrop mode.
	2	Field not displayed.	HP-IB network, with HP-IB printer or HP-IB plotter or both (at different HP-IB addresses).

The port for connection of external devices to the terminal is illustrated in figure 7-1.

Figure 7-1. External Device Connections



Plotter Connections

The source of control and data input for a plotter is a program operating on a host computer. Input consists of plotting commands (using the "Ec*p" escape sequence, such as Lift Pen, lower pen, etc.) embedded in the "Ec&p" data transfer escape sequences.

A plotter can be connected to the terminal in two connection types: eavesdrop and passthrough. The host computer program which controls the plotter must be aware of the type of plotter connection.

Eavesdrop Connection. In an eavesdrop connection (figure 7-1), plotter commands from the computer are screened out of the data stream by the plotter, so that they do not reach the terminal. An RS232C cable is used to connect the computer to the plotter. A second RS232C cable connects the plotter to the terminal's datacomm port.

In this connection, the terminal does not see the plotter as an external device.

Passthrough Connection. In a pass-through connection, the plotter commands go to the terminal, along with the rest of the data, where they are then transferred to the plotter.

An RS232C cable connects the computer to the terminal datacomm port (figure 7-1). A second cable connects the terminal external device port to the plotter. The second cable type depends on the option installed in port 2, and possibly on the "Remote/Serial Dev" field selection on the Global Configuration Menu.

In this connection, the terminal sees the plotter as an external device as concerns data transfer operations.

Cables

Table 7-4 lists the cables used to connect the terminal and external devices.

Table 7-4. External Device Cables

CABLE NO.	HP PART NO.	LENGTH	CABLE DESCRIPTION
40242A	40242-60008	1 m 3 ft	Male to female, RFI-protected, 25 to 50-pin converter cable.
40242C	40242-60003	5 m 16 ft	Male to female, RFI-protected, RS-232C, 25 to 25-pin printer cable.
40242G	40242-60006	5 m 16 ft	Male to male, RFI-protected, RS-232C, 25 to 25-pin printer cable.
40242M	40242-60004	5 m 16 ft	Male to male, RFI-protected, RS-232C, 25 to 25-pin modem cable. For connection to 202C-type European modems.
40242P	40242-60007	5 m 16 ft	Male to male, RFI-protected, RS-422, 25 to 5-pin cable for direct connection to HP 3000-series 44, 48, 64, and 68 systems with an ATP interface.
40242X	40242-60005	5 m 16 ft	Male to male, RFI-protected, RS-232C, 25 to 3-pin cable for direct connection to HP 3000-series 44, 48, 64, and 68 systems with an ATP interface.
40242Y	40242-60010	5 m 16 ft	Male to male, electromagnetic-protected, RS-232C, 25 to 25-pin cable for connection to the multi-plexer of an HP 1000, HP 2000, or HP 3000 system.
40242Z	40242-60002	5 m 16 ft	Male to female, RFI-protected, RS-232C, 25 to 25-pin modem bypass cable.
40242D	40242-60009	1 m 3 ft	Male to male, 36 to 36-pin, parallel-data (Centronix-type) printer cable.
45529A	8120-3445	1 m 3 ft	HP-IB connectors on both ends. Used for connecting external devices to the terminal in an HP-IB network.

Data Source Selection

For alphanumeric data transfers to an external device (usually a printer) originated at the keyboard, the terminal alphanumeric display is automatically selected as the data source. Such transfers are originated using the COPY LINE, COPY PAGE, or COPY ALL function keys.

When the [Graph copy] key is used to copy the graphics display to a raster-type printer, the graphics display is automatically selected as the data source.

For data transfers between devices and initiated by a program operating on a host computer, the data source is selected using the following escape sequence:

Ec&p <x>S	where <x>is
3	Alphanumeric display memory.
7	Graphics display memory.

Data Destination Selection

To control a device, you must select it as the destination device for device control commands. This can be done using either an escape sequence or the function keys.

From the Keyboard

To select the external device as the destination device using function keys, press the following keys, in sequence:

[System], device control, "to" devices, and TO EXT DEV.

An asterisk in the TO EXT DEV function key label indicates the external device is selected. Alternate presses of the key display and delete the asterisk.

To select the Alphanumeric display as the destination device using function keys, press the following keys, in sequence:

[System], **device control**, **"to" devices**, and **TO DISPLAY**.

An asterisk in the **TO DISPLAY** function key label indicates the alphanumeric display is selected. Alternate presses of the key display and delete the asterisk.

If no option is installed in port 2 of the terminal, the **device control** function key will not appear when the System function keys are displayed. Also, alphanumeric display memory will be selected, by default, as the destination device.

If any of the options 046, 092, or 093 is installed in the terminal, the **device control** function key is displayed with the System function keys. Also, at power up, the external device is selected, by default, as the destination device.

From a Program

The escape sequence to programmatically select the data destination is listed below. More than one destination device can be selected by including one or more "d" parameters in the escape sequence.

```
Ec&p <x> D
```

x	DEVICE
3	Alphanumeric display memory.
4, 5, or 6	The connected external device.

Printer Control

Two levels of function keys are provided for local support of external printers.

Selecting the Printer Protocol

The printer protocol is selected by entering either "`HP`" or "`Other`" in the "`Protocol`" field of the appropriate one of the three external device configuration menu. Depending on the selected protocol, the PRESET PRINTER, EXPAND PRINT, COMPRESS PRINT, REPORT PRINT, and METRIC PRINT function keys, when pressed, send different escape sequences to the printer.

For any HP printer, use HP protocol. For the Diablo C150 printer, enter "`Other`" in the "`Protocol`" field of the menu.

Presetting the Printer

The PRESET PRINTER key can be used to preset a printer to a known state by causing the terminal to send the printer a string of three escape sequences. The escape sequences sent depend on the protocol selected in the "`Protocol`" field of the menu.

HP PROTOCOL

`Ec E` — Performs a hard reset.

`Ec&l 01 00p 54F`

where:

- `01`— Sets Perforation Skip mode off.
- `00p`— Selects the default page length.
- `54F`— Selects 54 lines of text per page.

`Ec&s 0C` — Selects end of line wraparound.

	OTHER PROTOCOL
`Ec <CR>P`	Performs a hard reset.
`Ecl 5<CR>`	Sets the left margin at the fifth column.
`Ecr 85 <CR>`	Sets the right margin at the 85th column.

Data Logging Modes

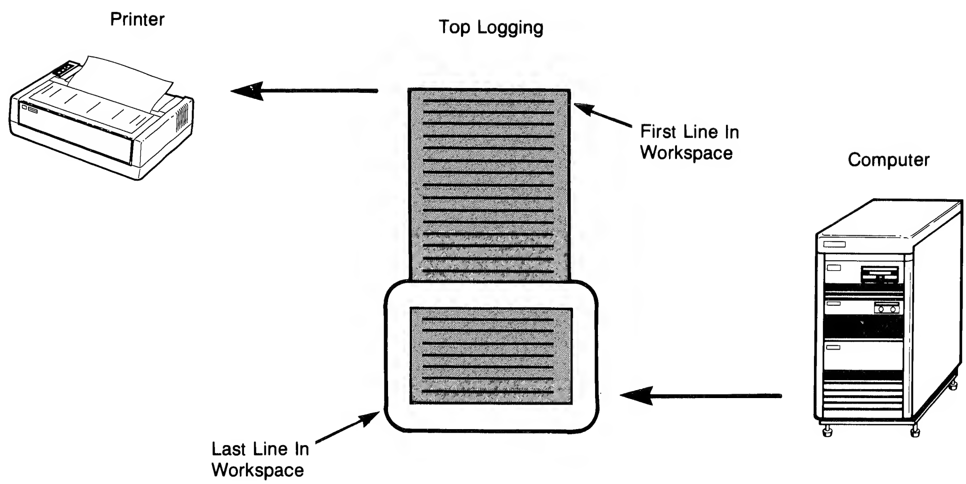
The terminal includes a feature called "data logging" whereby data arriving at the terminal from an external source can be automatically routed to the printer. There are two types of data logging: top and bottom.

Both forms of data logging cannot be enabled simultaneously. Once either form is enabled, it remains enabled until explicitly disabled, until the other form of data logging is enabled, until a hard reset is performed, or until the power is turned off.

The keyboard is temporarily locked while a line of data is being "logged". This may make it difficult to perform any keyboard operations if a large quantity of data is arriving over the datacomm line rapidly enough to result in continuous logging.

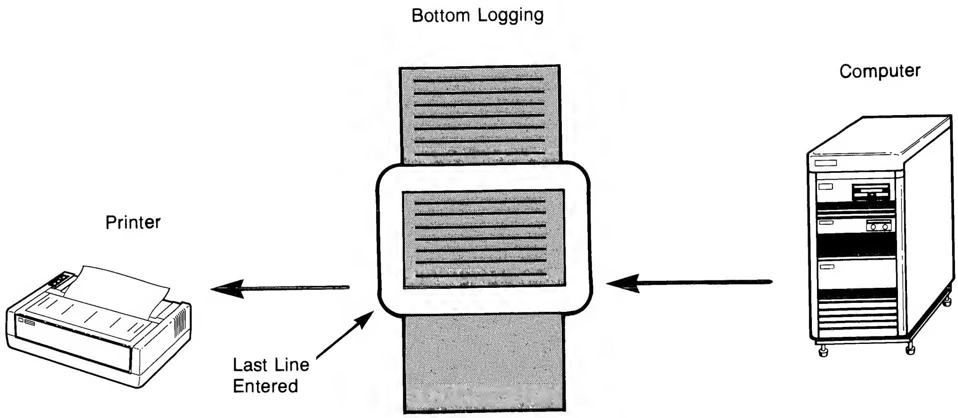
Log Top Mode. When the display is filled and another line of data is entered through the keyboard or received over a datacomm line, the top line in the display is purged to make room for the new line. With top logging, each line that is purged from the top of the display is printed. Thus, while the line is "lost" from display memory, it is maintained in hard copy form. Figure 7-2 illustrates top logging.

Figure 7-2. Top Logging



Log Bottom Mode. With bottom logging, each time the cursor moves from one line to another as the result of an explicit line feed or an end-of-line wraparound, the line from which the cursor moved is printed. This feature allows you to maintain a hard copy "trail" of all lines added to the display in the order in which they were entered and/or received. Figure 7-3 illustrates bottom logging.

Figure 7-3. Bottom Logging



Data Logging Handshaking. When performing data logging in Remote mode, the terminal and host computer must be using the `ENQ-ACK` or `XON-XOFF` handshakes, or they must be using a baud rate equal to or less than the rate at which the printer can function. (You will probably have to drop to 600 baud, if handshaking isn't used.)

Initiating and Ending Data Logging. From the keyboard, you enable and disable data logging using the LOG TOP and LOG BOTTOM keys. These keys alternately enable and disable top logging and bottom logging, respectively. When either is enabled, an asterisk appears in the associated key display.

From a program executing in a host computer, you enable and disable data logging using the following escape sequences:

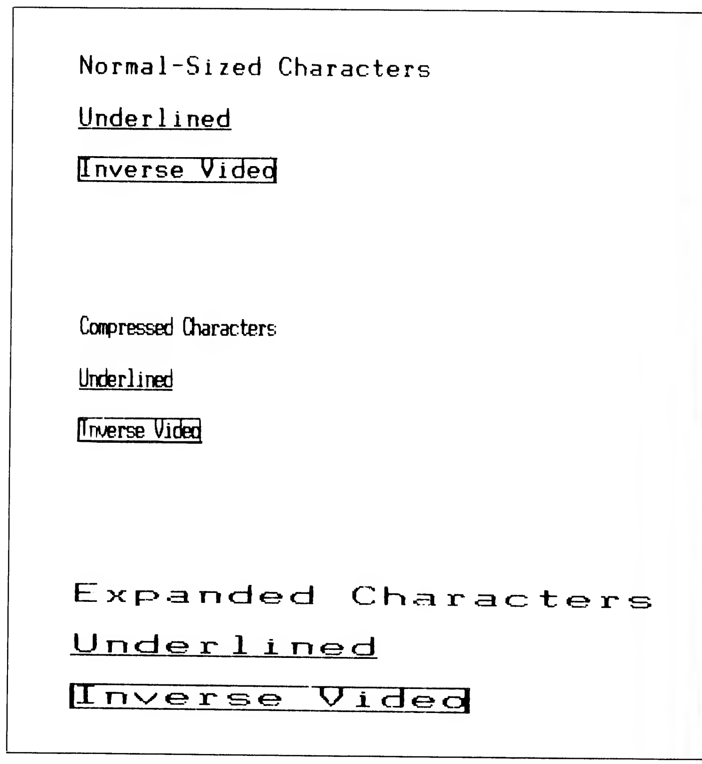
ENABLE BOTTOM LOGGING:	`Ec&p11C`
ENABLE TOP LOGGING:	`Ec&p12C`
DISABLE LOGGING:	`Ec&p13C`

Expand Mode

Expanded print can be produced on the printer from the terminal, provided the printer has an expand print capability which is triggered by an "Ec &k <x>S" sequence (figure 7-4).

Once the printing of expanded characters is enabled, it remains enabled until disabled, until Compress mode is enabled, until a hard reset is performed, or until the power is turned off.

Figure 7-4. Character Sizes and Enhancements as Printed on the Printer



From The Keyboard. From the keyboard, you enable and disable expanded character printing by pressing the EXPAND PRINT key. When enabled, an asterisk is present in the key label.

Action of the EXPAND PRINT key depends on the selection in the "`Protocol`" field of the appropriate external device menu. The key has no effect unless "`HP`" is selected as the protocol. With HP protocol selected, the enable or disable escape code is sent to the printer from the terminal when the key is pressed. The escape codes are the same as those sent from a program, as listed under the heading "From a Program".

From A Program. From a program operating in a host computer, you enable and disable the printing of expanded characters using the following escape sequences:

ENABLE: `Ec&k 1S`
DISABLE: `Ec&k 0S`

Compress Mode

Compressed print can be produced on the printer from the terminal, provided the printer has an expand print capability which is triggered by an "`Ec &k <x>s`" sequence (figure 7-4).

Once the printing of compressed characters is enabled, it remains enabled until disabled, until expanded characters are enabled, until a hard reset is performed, or until the power is turned off.

From The Keyboard. From the keyboard, you enable and disable compressed character printing using the COMPRESS PRINT key. When enabled, an asterisk is present in the key label.

Action of the COMPRESS PRINT key depends on the selection in the "Protocol" field of the appropriate external device menu. The key has no effect unless "HP" is selected as the protocol. With HP protocol selected, the enable or disable escape code is sent to the printer from the terminal when the key is pressed. The escape codes are the same as those sent from a program, as listed under the heading "From a Program".

From A Program. From a program executing in a host computer, you enable and disable the printing of compressed characters using the following escape sequences:

ENABLE: `Ec&k 2S`
DISABLE: `Ec&k 0S`

Record Mode

In Record mode, data is copied from the display (in Local mode) or from the computer (in Remote mode) to the selected "to" devices.

From The Keyboard. To activate Record mode using the function keys, press the following keys, in sequence:

[System], device control, device modes, RECORD MODE

In Local mode, pressing the RECORD MODE function key acts as a one time event, and is equivalent to pressing the [Print] key or the [Shift] and [Enter] keys together.

In Remote mode, pressing the RECORD MODE function key puts the terminal in Record mode, where it remains until disabled. If the alphanumeric display is selected as the destination device, the data stream from the computer is stored in terminal memory and displayed as it is stored.

An asterisk is present in the RECORD MODE key label when Record mode is active. Alternate presses of the key display and delete the asterisk. While in Record mode, the keyboard is disabled, except for the [Break], [Reset], and RECORD MODE keys. The keyboard is disabled after one key press.

From A Program. To initiate Record mode programmatically, send the terminal the following escape sequence:

```
Ec&p <char>p 20C
```

where: the optional parameter "<char>p" is the decimal equivalent of an ASCII character which can be used to turn off Record mode. It should be the first character in a record; the default character is "0". If "<char>p" is omitted or if "0p" is specified, no character will turn off Record mode. Termination can only occur by pressing RECORD MODE or initiating a soft or hard reset from the keyboard.

The termination character selected in the escape code is valid only for the current activation of Record mode. When Record mode is ended, the termination character returns to the default character ("0").

If the escape sequence is received from the computer, the terminal returns an "S" or "F" character to the program to indicate "successful" or "failed" status execution of the escape sequence. This status check is discussed later in this section.

Whether or not the DC1/DC2 handshake is enabled determines when the status character is sent. (Refer to Section 10 for information on handshake types.) If the DC1/DC2 handshake is disabled, the character (always an "S") is sent immediately after the terminal receives the escape sequence; otherwise, it (an "S" or "F") is sent after Record mode is turned off and a DC1 is received from the computer.

A 256-character buffer is used to hold each record prior to sending it to the "to" device(s). If the record exceeds 256 characters, the terminals handshake holds off any further transmission from the computer until the buffer contents are sent to the "to" device(s). Records shorter than 256 characters are indicated by an LF (line feed) character. In this case also, the terminals handshake holds off further transmission from the computer until the buffer is cleared.

If Record mode is turned off with a partially-filled buffer, the contents are sent to the "to" device(s).

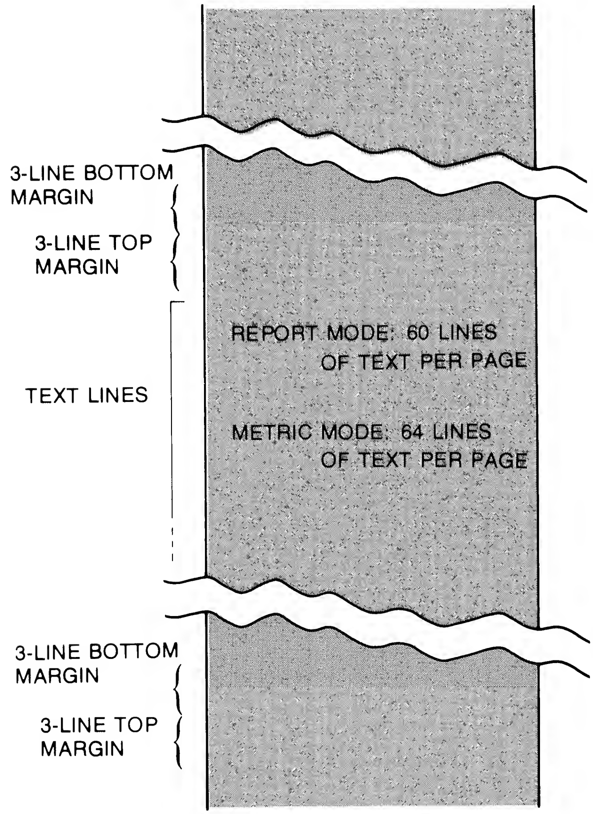
If the first character in the buffer is the termination character, Record mode is terminated and the termination character is not sent to the "to" device(s).

Report Mode

You can enable Report mode in which printed output is treated as a series of 66-line pages (a 3-line top margin, 60 lines of text, and a 3-line bottom margin). The margins and text area together form an 8-1/2 inch by 11-inch page. Some printers use a small tic mark to mark the end of one page and the beginning of the next. Report mode output is shown in figure 7-5.

Once enabled, Report mode remains enabled until disabled, until Metric Report mode is enabled, until a hard reset is performed, or until the power is turned off.

Figure 7-5. Report and Metric Report Formats



From the Keyboard. From the keyboard, you enable and disable Report mode using the REPORT PRINT key. This key alternately anables and disables Report mode. When enabled, an asterisk appears in the key label (indicating the mode is active); Metric mode, if on, is turned off.

Action of the REPORT MODE key depends on the selection in the "`Protocol`" field of the appropriate external device menu.

With HP protocol selected,the enable or disable escape code is sent to the printer from the terminal when the key is pressed.

ENABLE: `Ec&l 1l 66p 60F`

where: `1l` —Sets Perforation Skip mode.

`66p` —Selects a page length of 66 lines.

`60F` —Selects 60 text lines per page.

DISABLE: `Ec&l 0l 00p 54F`

where: `0l` —Disables Perforation Skip mode.

`00p` —Selects the default page size.

`54F` —Selects 54 text lines per page.

When "`Other`" is selected in the "`Protocol`" field of the menu, the terminal enables and disables Report mode with the following escape sequences:

ENABLE: `Ec <FF> 60` (Selects 60 lines per page.)

DISABLE: `Ec <FF> 54` (Selects 54 lines per page.)

where `<FF>` is the form feed character.

From A Program. From a program executing in a host computer, you enable and disable Report mode using the following escape sequences:

ENABLE: `Ec&p 17C`

DISABLE: `Ec&p 19C`

Metric Report Mode

You can enable Metric Report mode in which printed output is treated as a series of 70-line pages (a 3-line top margin, 64 lines of text, and a 3-line bottom margin).

Once enabled, Metric Report mode remains enabled until disabled, until Report mode is enabled, until a hard reset is performed, or until the power is turned off.

From the Keyboard. From the keyboard, you enable and disable Metric Report mode using the METRIC PRINT key. This key alternately anables and disables Metric Report mode. When enabled, an asterisk appears in the key display; Report mode, if on, is turned off.

Action of the METRIC PRINT key depends on the selection in the "`Protocol`" field of the appropriate external device menu.

With HP protocol selected, the enable or disable escape sequence is sent to the printer when the key is pressed.

ENABLE: `Ec&l 1l 70p 64F`

where: `1l` —Sets Perforation Skip mode.

`70p` —Selects a page length of 70 lines.

`64F` —Selects 64 text lines per page.

DISABLE: `Ec&l 01 00p 54F`

where: `01` —Disables Perforation Skip mode.

`00p` —Selects the default page size.

`54F` —Selects 54 text lines per page.

When "`Other`" is selected in the "`Protocol`" field of the menu, the terminal enables and disables Report mode with the following escape sequences:

ENABLE: `Ec <FF> 64` (Selects 64 lines per page.)

DISABLE: `Ec <FF> 54` (Selects 54 lines per page.)

where `<FF>` is the form feed charact

From a program executing in a host computer, you enable and disable Metric Report using the following escape sequences:

ENABLE: `Ec&p 18C`

DISABLE: `Ec&p 19C`

Paper Movement

Two paper movement operations are allowed: line feed (advance line) and form feed (advance page).

To perform one of these operations using the function keys, press the System and device control keys, in sequence. Then press either the ADVANCE LINE or ADVANCE PAGE key, as desired. Pressing the ADVANCE LINE key sends an ASCII `<CR> <LF>` control code sequence to the printer. Pressing the ADVANCE PAGE key sends an ASCII `<FF>` control code to the printer.

Programmatically, you can produce a printer line feed by sending the following escape sequence:

```
Ec&p 4u 1p 1C
```

The number preceding the parameter "p", in the escape sequence, specifies how many line feeds you want. For example, to generate four successive line feeds, preceed the "p" with the number "4". Refer to Appendix A for a complete explanation of this escape sequence.

To produce a form feed from a program, send the terminal the following escape sequence:

```
Ec&p 0c 4U
```

(The values 2 through 10, preceding the "c" in the escape sequence, will also produce a form feed.)

HP-IB Network Control

You can operate simultaneously an HP-IB printer and an HP-IB plotter connected to the terminal with option 046 on port 2. The printer would normally be at address 01 and the plotter would typically be at address 05. This allows local data transfers to the printer using the function keys and graphics keypad.

A host program, such as HPDRAW, can be used to enable the user to plot directly, by sending the following commands to a plotter in an HP-IB network.

`Ec&p <x>p 5u <z>C` where:

<z>	ACTION
1	Use <x> as the device address for subsequent bus operations, normally on a "talker" device.
2	Use <x> as the device address for subsequent bus operations, normally on a "listener" device.
3	Enable HP-IB device operations timeout.
4	Disable HP-IB device operations timeout.

Data Transfer: Alphanumeric Display to Printer

You can copy from the display a selected line, a part (or all) of the displayed page, or a part (or all) of the active workspace. The cursor is used as the selector to determine the line at which the copy operation starts. To perform these operations, you must select the printer as the destination ("to") device. When the print operation is initiated from the keyboard, the display is automatically defined as the "from" device.

The operation sequence is to select the printer as a "to" device, place the cursor in the line at which you want printing to start, then display the "device control" set of function keys and select COPY LINE, COPY PAGE, or COPY ALL.

Copy Line

When the printer is selected as a destination device, you can copy the line containing the cursor from the display to the printer. The entire line is copied. Block terminators are ignored. After the line is printed, the cursor moves to the leftmost column in the next lower line (column 0, not the left margin). If the cursor is on an empty line, COPY LINE should not cause anything to be printed.

From the keyboard, you copy one line of data using the COPY LINE key in the "device control" set of function keys.

From a program executing in a host computer, you copy one line of data using the following escape sequence:

```
Ec&pB
```

Copy Page

When the printer is selected as a destination device, you can copy all lines, starting with the line containing the cursor through the last line visible on the screen, to the printer. Block terminators are ignored. After each line is printed, the cursor moves to the leftmost column in the next lower line (column 0, not the left margin). If the cursor is at a line beyond the last displayable line, the printer does nothing.

From the keyboard, you copy a "page" of data using the COPY PAGE key in the "device control" set of function keys.

From a program operating in a host computer, you copy a page of data using the following escape sequence:

```
Ec&pF
```

Copy All

When the printer is selected as a destination device, you can copy all lines, starting with the line containing the cursor through the last line of display memory, to the printer. Block terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 0, not the left margin). If the cursor is located beyond the last displayable line, the printer does nothing.

From the keyboard, you copy "all" using the COPY ALL key in the "device control" set of function keys.

From a program executing in a host computer, you copy "all" using the following escape sequence:

```
Ec&pM
```

Copy the Entire Alphanumeric Memory

When the printer is selected as a destination device, you can copy the entire content of the alphanumeric memory to the printer by "homing up" the cursor and selecting the COPY ALL key. Block terminators are ignored.

From a program executing in a host computer, you copy the entire alphanumeric memory using the following escape sequence:

```
Ec0
```

The entire alphanumeric memory, not just that portion which appears on the screen, is copied from the display to the printer.

Data Transfer: Computer To External Device

With the external device selected as a destination device, you can transfer upto 256 characters of a data string from the computer to the external device. The data can be sent in either binary or ASCII form, depending on the escape sequence used.

Binary Data Transfer

If data is to be transmitted to a printer (or other external device) in binary form, the device must be of a type which accepts binary data. Also, the ENQ/ACKform of handshaking must be used. This means that the computer program which initiates the data transfer must transmit an ENQ ("5" in ASCII decimal code) immediately before transmitting the data.

The following escape sequence is used to initiate transfer of a string of binary data:

```
Ec&p <x>W <data string>
```

This sequence transfers the first "x" bytes of the data string, in binary form, to the printer. The maximum value for "x" is 256.

The sequence of events for transmitting five bytes of binary data ("12345") to an external device from a program, using the ENQ/ACK handshake, is shown below:

```
COMPUTER PROGRAM          TERMINAL

        Ec&p5W -------->
           ENQ -------->
               <-------- ACK
       "12345"  ------->
```

As an alternative to the ENQ/ACK handshake, the program can pause for 3 seconds after sending the initiating escape code.

ASCII Data Transfer

To copy an ASCII data string from the computer to the printer, use the following escape sequence:

```
Ec&p W <data string>
```

The entire data string is copied. The string is terminated either by an ASCII line feed character or by the 256th character.

Data Transfer: Graphics Display To Printer

You can obtain a hardcopy of the screen display using either a printer or plotter. However, a plotter copy can only be obtained through use of a program, such as HPDRAW, which stores the picture data as it is entered at the terminal and transfers it to the plotter, after performing the necessary interface operations.

With a suitable printer, capable of accepting raster dump data, connected to the external device port, you can obtain a copy of the contents of raster memory using the keyboard or a program which requests a raster dump.

You control the size, layout, and content of the copy by configuring the appropriate configuration menu. The External Parallel Device Configuration menu is for parallel-input devices, and the External HP-IB Device Configuration menu is used for HP-IB-connected devices. Refer to Section 6 for configuration details.

From the Keyboard

To copy raster memory to a printer, the printer must be selected as the "to" device. To do so, press the [System], **device control**, **"to" devices**, and **TO EXT DEV** keys, in sequence.

The [Graph copy] key on the graphics/numeric pad initiates the graphics transfer when the keypad is in graphics mode. [Shift] and [NUM], pressed together, toggle the function of the keypad between graphics mode and numeric mode. Pressing the [Enter] key on the graphics keypad, or simultaneously pressing the [CTRL] and [Enter] keys on the alphanumeric keypad will also perform the copy operation. The current state, numeric or graphics, is displayed in the status line at the bottom of the display.

From a Program

Coding a program to transfer graphics data to the printer requires selecting the raster memory as the data source and the external printer as the destination. The escape sequence is:

```
Ec&p 7s <x>d
```

where: `7s` Selects raster memory as the source.

`<x>d` Selects the external device (printer, plotter, or HP-IB network) as the destination. `<x>` can be any of the numbers 4, 5, or 6.

Example: Define raster memory as the source and the printer, connected as the external device,as the destination.

```
Ec & p 7s 4D
```

After the source and destination are defined, the transfer is initiated by either:

```
Ec & p F
```

or

```
Ec & p M
```

You may combine the source and destination assignments and the transfer initiation in one escape sequence:

```
Ec & 7s 4d F
```

Determining If Your Escape Sequence Command Has Been Successfully Performed

After issuing a copy line, copy page, copy all, advance line, or advance page `Ec&p` sequence, the remote program can determine whether or not the operation was successfully performed by executing an INPUT (BASIC language) or similar instruction that requests one ASCII character from the terminal.

The terminal responds by sending an "`S`", "`F`", or "`U`". An "`S`" indicates successful completion, an "`F`" indicates that the operation failed, and a "`U`" indicates that the terminal operator interrupted the data transfer by pressing [Return]. A "`U`" can be returned only for a copy page or copy all operation.

If the printer is offline when an attempt is made to transfer data to it, the terminal will still return an "`S`".

These completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands for accepting them.

The keyboard is disabled ("locked") until the status is sent.

In either Character or Block Line mode, the terminal sends a `<CR>` (or a `<CR><LF>` if Auto Linefeed mode is enabled) following the completion code. In Block Page mode, it sends a block terminator character.

If a datacomm error occurs during transmission of the data record, the device control completion code is unpredictable. Datacomm errors are reported by way of the terminal status bytes described in Section 9.

8 Data Communications

Introduction

Data communications (datacomm) refers to the transfer of data between the terminal and a host computer.

The following items are discussed in this section:

- Considerations involved in selecting either a modem or hardwired connection.
- Installation.
- Configuration.
- Information required for writing programs which interface with the datacomm system.

This terminal is an asynchronous, full-duplex terminal, which must be connected in a point-to-point configuration.

Following is a list of datacomm terms with their definitions, as used in this discussion.

Data Link	The means by which a terminal is connected to a host computer. This always includes some type of communications line (a coaxial cable, the public telephone network, or a leased telephone line), and it may also include a pair of modems (one at each end of the line).
Asynchronous	A mode of transmission in which each data character is framed by a "start bit" and one or more "stop bits". The interval between successive data characters is random.

Synchronous	A mode of transmission in which data is sent in a continuous stream with no intervals between successive characters. When there is no data being sent, the communications line is in the "idle" or "ones" state. At the start of, and during, each transmission the terminal and the computer maintain synchronization with one another through the use of SYN (ASCII decimal code 22) control characters.
Full Duplex	A data link in which data can be transmitted in both directions simultaneously.
Half Duplex	A data link in which data can be transmitted in only one direction at a time. Each time the direction of the data flow is reversed, the modems on each end of the line must switch from "transmit" state to "receive" state (or vice versa). This state transition is called a "line turnaround".
Point-to-Point	A data communications configuration in which a single terminal is connected to a host computer over a data link.
Multipoint	A data communications configuration in which two or more terminals are "chained" together so as to share a data link to a host computer.
Character Mode	When the terminal is operating in Character mode, it sends data characters to the computer one at a time as they are typed into the keyboard.

Block Mode

When the terminal is operating in Block mode, data characters typed into the keyboard are merely stored in display memory. When a block transfer is subsequently triggered (by the host computer or by pressing the Enter key or another appropriately defined key), a group of data characters is sent from the terminal to the computer as a block.

Terminal Datacomm Capabilities

The terminal datacomm capabilities are listed in table 8-1.

Table 8-1. Terminal Datacomm Capabilities

TERMINAL	DATACOMM FEATURES
Standard Terminal	Port 1, the only port, can be used for either an RS-232-C or a direct-connect RS-422 datacomm connection.
Option 092	Port 1 can be used for either an RS-232-C or a direct-connect RS-422 datacomm connection. Port 2 is an auxilliary RS-232-C datacomm port. Either port 1 or port 2 can be used as the datacomm port.
Option 093	Port 1, the only datacomm port, can be used for either an RS-232-C or a direct-connect RS-422 datacomm port. Port 2 is a Centronics-type, parallel-data printer port.
Option 046	Port 1, the only datacomm port, can be used for either an RS-232-C or a direct-connect RS-422 datacomm connection. Port 2 is an HP-IB port.

Connection Considerations

For each desired data link you must decide whether you want a hardwired or modem connection.

A hardwired connection, where feasible, is the cheaper alternative because it eliminates the use of modems and common carrier (telephone company) lines.

A major consideration in selecting which type of connection to use is the anticipated distance between the terminal and the computer. If the terminal will be located in the vicinity of the computer system you may use a hardwired connection. RS-232-C specifications limit cable lengths to a maximum of 50 feet (15 meters); RS-422 specifies cable lengths from 200 to 4,000 feet (60 to 1,220 meters).

Another consideration is the desired availability of the particular computer port. If you wish to have it available (at different times) to terminals in diverse and/or varying locations, then you should choose a modem connection with dial-up capability.

Hardwired Connections

If you have chosen a hardwired connection, all that remains is to select the cable. The available cables are covered later.

Modem Connections

If you have chosen a modem connection you must now decide what type of modem to get. Since terminals connected point-to-point always employ asynchronous transmission, your choice of modem must be one of the asynchronous variety. The modem selection is listed in table 8-2.

Table 8-2. Modems

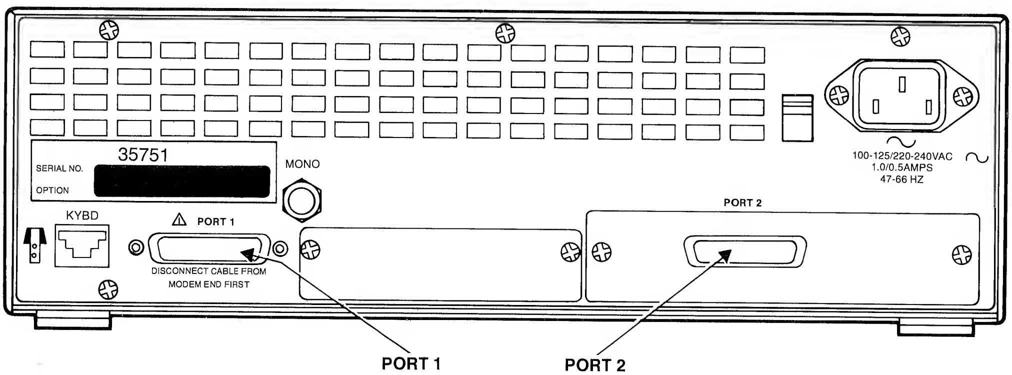
MODEM	DATA RATE (BITS/SEC)	DUPLEX FULL/HALF	DIALED/ LEASED
Bell 103A	300	F/H	D/L
Bell 202T Bell 202D	1200 (see note 2)	F/H	D/L
Bell 212A	300 or 1200	F	D
Vadic VA3400 (see notes 1, 3)	1200	F	D

Notes:
1. Can be configured for either asynchronous or synchronous operation. Must be set for asynchronous operation for use with this terminal.
2. C2 line conditioning allows operation at 1800 bits per second.
3. Must include the internal clock option.

Installation

On the standard terminal and all options except Option 092, port 1 is the datacomm port. On Option 092, either port 1 or port 2, or both, can be used as a datacomm port. Either port can also be used for connection to a peripheral device. Both ports are located on the rear panel (figure 8-1).

Figure 8-1. Terminal, Rear View (Typical)



Cabling

Port 1 is normally used for connecting the terminal to a host computer; however, for terminals with the RS-232 second port option (Option 092) port 2 can also be used to connect to a host computer. Cable connection of the terminal to a host computer is illustrated in figure 8-2. Cables useable for connecting the terminal to a host computer are listed in table 8-3.

Figure 8-2. Cabling Connections

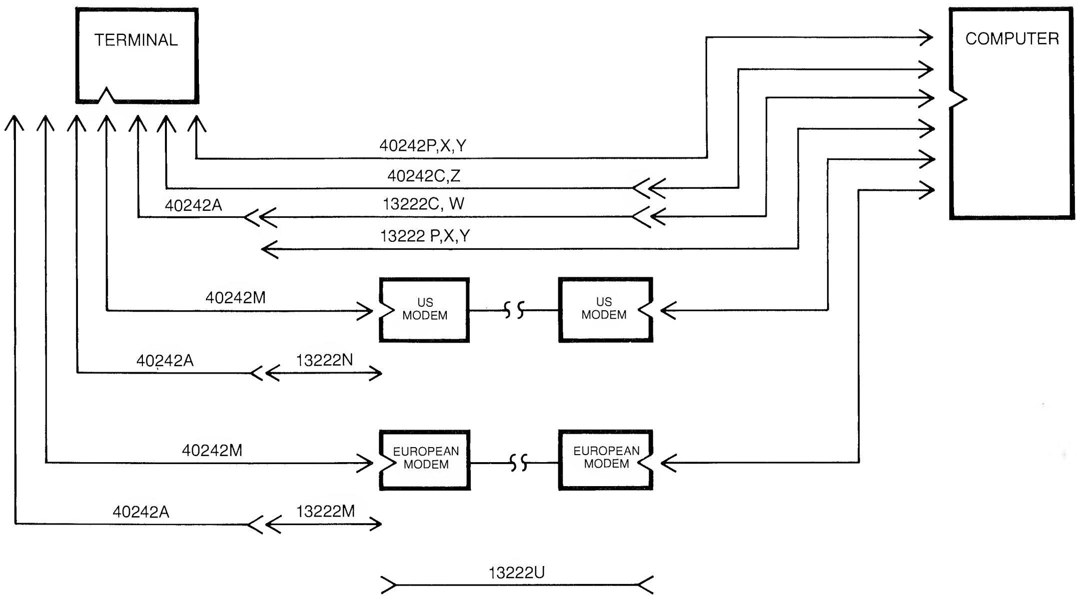


Table 8-3 provides a brief description of the cables useable for configuration.

Table 8-3. Point-to-Point Cables

CABLE NO.	HP PART NO.	DESCRIPTION
40242A	40242-60008	Combined RS-232-C and RS 422 adapter cable, male 25 pin to female 50 pin. Provides for 50-pin connections to computer. Length: 1 meter (3 feet).
40242C	40242-60003	RS232C cable, male 25 pin to female 25 pin. Length: 5 meters (16 feet).

Table 8-3. Point-to-Point Cables (continued)

CABLE NO.	HP PART NO.	DESCRIPTION
40242M	40242-60004	US/European modem cable, male 25 pin to male 25 pin. Male RS232C connector cable for interfacing the terminal to the European telephone system via a Bell 103 or 202C type European modem. Length: 5 meters (16 feet).
40242P	40242-60007	HP direct-connect type 422 cable, male 25 pin to male 25 pin. Length: 5 meters (16 feet).
40242X	40242-60005	HP direct-connect type 232 cable, male 25 pin to male 3 pin. Used to interface the terminal to an HP 3000 Series 44 or 64. Length: 5 meters (16 feet).
40242Y	40242-60010	EMP protect cable, male 25 pin to male 25 pin. RS232C connector for interfacing the terminal to an HP 1000 or 3000 multiplexer. Provides protection from lightning-induced transients. For use in hardwired cofigurations only. Length: 5 meters (16 feet).
40242Z	40242-60002	RS232C modem bypass cable, male 25 pin to female 25 pin. RS232C direct-connect cable. Length: 5 meters (16 feet).

Table 8-3. Point-to-Point Cables (continued)

CABLE NO.	HP PART NO.	DESCRIPTION
13222C	13222-60003	Terminal to RS232 cable (used with 40242A cable), male 50 pin to female 25 pin. Female RS232C 25-pin connector. Length: 2 meters (6.6 feet).
13222M	13222-60002	European modem cable (used with 40242A cable), male 50 pin to male 25 pin. Male RS232C 25-pin connector for interfacing the terminal to the European telephone system via Bell 103 or 202C type European modems. Length: 5 meters (16 feet).
13222N	13222-60001	US modem cable (used with 40242A cable), male 50 pin to male 25 pin. Male RS232C 25-pin connector for interfacing the terminal to an HP 1000 or 3000 multiplexer; to a Bell 10MA, 202C/D/S/T, 212A, or Vadic 3400 modem; or to an acoustic coupler (signal compatible only). Length: 5 meters (16 feet).
13222P	13222-60009	HP direct-connect type 422 cable (used with 40242A cable), male 50 pin to male 5 pin. Length: 5 meters (16 feet).

Table 8-3. Point-to-Point Cables (continued)

CABLE NO.	HP PART NO.	DESCRIPTION
13222W	13222-60007	RS232C cable (used with 40242A cable), male 50 pin to female 25 pin. Female RS232C 25-pin connector for interfacing the terminal to an HP 3000 computer system. Length: 5 meters (16 feet).
13222X	13222-60008	HP direct-connect type 232 cable (used with 40242A cable), male 50 pin to male 3 pin. Length: 5 meters (16 feet).
13222Y	13222-60005	EMP protect (used with 40242A cable) cable, male 50 pin to male 25 pin. Male RS32C 25-pin connector for interfacing the terminal to an HP 1000 or 3000 multiplexer. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 5 meters (16 feet).
13232U	5061-2403	Modem bypass cable, female 25 pin to female 25 pin. Crosses the signals so that two terminals can communicate with one another. Length: 1.5 meters (5 feet).

Configuration

Refer to Section 6 for configuration instructions.

Programming Information

The following paragraphs supply programming information of interest to someone writing a data communications driver or controller program to communicate with this terminal in an asynchronous point-to-point environment. An asynchronous point-to-point data communications environment is characterized by a flow of characters that have been produced over random time intervals. To achieve hardware synchronization, each character is delimited by a "start bit" and one or more "stop bits".

Character Mode Operation

When the terminal is configured for Character mode operation (Block mode disabled), the terminal sends characters to the host computer as they are entered through the keyboard. This mode of operation can be used for interactive or conversational exchanges between the terminal operator and an application program.

Block Mode Operation

When the terminal is configured for Block mode operation (Block mode enabled), data entered through the keyboard is queued by the terminal and sent as a block after the [Enter] key is pressed. If handshaking is disabled, the data block is sent when the [Enter] key is pressed. When the DC1/DC2/DC1 handshake is enabled, pressing the [Enter] key causes the terminal to send a DC2 to the host computer after a DC1 is received and then send the data block when the computer responds with another DC1. The operation of the [Enter] key is described in detail in Section 10.

There are certain other functions which always result in a multicharacter (block) data transfer.

- Terminal-to-computer data transfers initiated by an "Ec&p" or "Ec d" sequence.
- User key-to-computer data transfer ("T" type)
- Responses to status requests from the host computer.
- Responses to cursor sensing requests from the host computer.

The driver program at the host computer must support whatever handshaking process is configured at the terminal (no handshake, DC1 trigger handshake, or DC1/DC2/DC1 handshake). In the latter case, the DC2 must be recognized as a request to send data and the DC1 must be sent to trigger the transfer after system buffers have been allocated to receive the data block. Additional software support may be needed depending upon your need for terminal or device control. The InhHndShk(G) and InhDC2(H) fields of the Terminal Configuration menu specify which form of handshaking the terminal will use. The Terminal Configuration menu is described in Section 6. For more detailed information on handshaking, refer to Section 10.

The driver program must also support the type of datacomm handshaking selected on the Remote Datacomm Configuration menu "XmitPace" and "RecvPace" fields. Refer to "Transmit Pacing Mechanisms" and "Receive Pacing Mechanisms", later in this section, for further information.

The operation of multicharacter transfers is described in Section 10.

Note The computer should not be allowed to echo back information that has been transmitted as a block from the terminal.

Start and Stop Bits

These hardware-generated bits are used for synchronizing the transmit and receive devices in an asynchronous environment. A start bit is a "zero" line state that lasts for 1.0 bit time; it is affixed to the beginning of a serial character-bit stream (which may also include a parity bit). A stop bit is a mark or a "one" line state that lasts for 1.0, or 2.0 bit times; it is appended to the end of each serial character-bit stream. After the stop bit, the line remains in the mark state until the next character, signified by a start bit, is transmitted.

Parity Checking

This terminal offers the following five types of parity, selected on the "`Parity/DataBits`" field of the Remote Datacomm Configuration menu.

1. `0'S/7`. The high-order bit is always a 0 (space).
2. `1'S/7`. The high-order bit is always a 1.
3. `ODD/7`. The high-order bit is set to a 0 or a 1, whichever produces an odd number of one bits in the character representation (the seven data bits plus the eighth parity bit).
4. `EVEN/7`. The high-order bit is set to a 0 or a 1, whichever produces an even number of 1 bits in the character representation (the seven data bits plus the eighth parity bit).
5. `NONE/8`. No parity bit is sent, all eight bits are significant data bits.

Receive Buffer

The terminal's receive buffer is a first in/first out (FIFO) storage area for accepting data from the remote device. When you are using any type of receive pacing (any selection except "`None`" in the "`RecvPace`" field of the Remote Datacomm Configuration menu), the buffer is partitioned into a working buffer and a 63-byte overrun area. The buffer size is 255 bytes, thus if receive pacing is being used, the working buffer is 192 bytes long and the overrun area is 63 bytes long.

If the received data overflows the working buffer and intrudes on the overrun area, the terminal will exercise whatever receive pacing mechanism is currently enabled (send an XOFF, for example, if XON/XOFF receive pacing is enabled) to temporarily halt the flow of data from the remote device. When enough data has been processed so that the receive buffer is only one quarter full (64 bytes), the terminal signals the remote device to resume transmission (by sending an XON, for example, if XON/XOFF receive pacing is enabled).

Receive Errors

When receiving data from the remote device, the terminal can detect the following three types of error conditions (in addition to parity errors):

1. Character overruns—a character is received before the preceding character was processed by the terminals datacomm firmware.
2. Framing errors—no stop bit was detected at the end of a character.
3. Buffer overflows—the entire allocated buffer space is filled (both the working buffer and the overrun area). The last character in the buffer will be overwritten by a "DEL" character. (If the remote device is using the selected form of pacing, this condition should never occur.)

Receive errors, when detected, cause a "DEL" character to be displayed on the screen at the point of the error. The host may determine if a datacomm error has occurred by inspecting byte 5 of the primary terminal status bytes (refer to Section 9, Status, for information on terminal status). The host computer will not be able to determine which type of error occurred.

Local/Remote Modes

The data communications portion of the terminal operates independently of Remote and Local modes. If the terminal is switched from Remote to Local while data is being received from the remote device, the datacomm portion of the terminal continues receiving data, and storing it in the buffer. In such a case, any data received in Local mode which overflows the buffer is discarded by the terminals firmware. Then, when the terminal reenters Remote mode, the data stored in the buffer in Local mode will be processed and sent to the screen. (To prevent buffer overflow errors when you switch from Remote to Local mode to stop data from being transferred to the screen, XON/XOFF receive pacing should be used.)

Full-Duplex Operation

This terminal is capable of transmitting and receiving data simultaneously. The ability to transmit may be inhibited temporarily, but it is never exclusive of the ability to receive. Two physical sets of data lines are required. Control lines are needed only when hardware handshaking or a modem is used.

When the terminal is connected to the host computer via a modem, the following primary control lines are required:

Request to Send (RS/CA)

Clear to Send (CS/CB)

Data Terminal Ready (TR/CD)

Data Mode (DM/CC)

Receiver Ready (RR/CF)

Transmit Pacing Mechanisms

This terminal can use any of the following forms of transmit pacing:

1. Hardware handshake. The host computer can temporarily restrain the terminal from transmitting by any one of the following means:

   a. Lowering the Clear to Send (CS/CB) line.

   b. Lowering the Secondary Receiver Ready (SRR/SCF) line (useable only on an external device.)

   c. Lowering the Data Mode (DM/CC) line.

   d. All of the above simultaneously.

   This type of transmit pacing can only be used in a hardwired configuration.

2. XON-XOFF handshake. The host computer uses the ASCII control codes XON (DC1) and XOFF(DC3) to start and stop the terminal from transmitting. A single XON code cancels any number of preceding XOFF codes.

Receive Pacing Mechanisms

This terminal can use any of the following forms of receive pacing:

1. Terminal Ready Pacing. The terminal can temporarily restrain the host computer from transmitting by lowering the Data Terminal Ready (TR/CD) line. It does this when its receive "working" buffer is full. When enough data has been processed so that the receive "working" buffer is only one quarter full, the terminal restarts transmission from the host by raising the TR/CD line.

   This type of receive pacing can only be used in a hardwired configuration.

2. Receiver Ready Pacing. When the Receiver Ready line is lowered, the terminal does not interpret received characters as data; it discards them. (This form of pacing cannot be used on terminals with an RS232C port on port 2—Option 092.)

3. XON-XOFF Pacing. The terminal uses the ASCII control codes XON (DC1) and XOFF (DC2) to start and stop host computer transmission. A single XON code cancels any number of XOFF codes.

4. ENQ-ACK Handshake. This is a Hewlett-Packard handshaking mechanism in which the host computer transmits a block of data and then sends an ASCII ENQ control code. The terminal responds to the ENQ by sending back an ASCII ACK control code when it has processed all of the data preceding the ENQ. The general interpretation of these two control codes is as follows:

   ENQ: "Have you processed the data up to this point?"

   ACK: "Yes, I have."

   If the host computer is an HP 1000 or HP 3000, it does not send any data following the ENQ until it has received the ACK, or until a timeout period (several seconds) has elapsed.

Pacing Mechanism Precedence

The above pacing mechanisms (both transmit and receive) are responded to by the terminal in the following order of precedence:

1. Hardware handshaking pacing (highest priority)
2. XON/XOFF receive pacing
3. XON/XOFF transmit pacing
4. ENQ/ACK pacing (lowest priority)

Note

If both XON/XOFF transmit pacing and XON/XOFF receive pacing are enabled, the receive pacing has priority, so that if the host computer sends XOFF, followed by data, the terminal can still respond with an XOFF before its buffer overflows. This algorithm should also be used by the host computer, as the terminal may send XOFF and follow it with transmit data. If both parties function in this way, then deadlock is prevented, and both parties should prevent buffer overrun at all times.

Stopping and Starting Computer Data Transmission

Transmission of data from the host computer to the terminal can be stopped and restarted from the keyboard by pressing the [Stop] key. This key toggles between stopping and restarting data transmission. It does so by sending a DC3 (XOFF—stop) or DC1 (XON—resume) or by withholding the next ACK character, and sending it when [Stop] is pressed again.

Simultaneously pressing [CTRL] and [S] also stops transmission by sending a DC3, and [CTRL] [Q] resumes transmission by sending a DC1.

Halting Terminal Data Transmission Temporarily

From a user key or from a program executing in a host computer, you can cause the terminal to pause for approximately 1 second using the following escape sequence:

```
Ec @
```

Multiple uses of this escape sequence in succession can be used to obtain virtually any desired time delay.

Note that while an `Ec @` is in effect, the cursor disappears from the screen, the keyboard is locked, and the passing of data from the datacomm firmware to display memory is inhibited.

For example, if you want to sound the bell tone twice in succession with a two-second delay between tones, you could do so using the following control sequence:

```
<BELL> Ec @ Ec @ <BELL>
```

Modem Disconnect

You can direct the terminal to "hang up" the modem by sending an `Ec f`. The terminal accomplishes the modem disconnect by lowering the TR/CD (Terminal Ready) line for 2 seconds.

9 Status

Introduction

Status requests are issued as escape sequences. On receipt of a status request, the terminal sends a block of status data to the computer. This data is in the form of a data block, such as is generated in Block mode. For additional information on transfer of block data, refer to Section 10.

There are several types of status requests:

1. Terminal Identification. This request is the means by which your program determines the kind of terminal it is communicating with.
2. Terminal Capabilities. There are four types of terminal capability requests:

    a. Alphanumeric capabilities.

    b. Graphics capabilities.

    c. Interface capabilities.

    d. Amount of RAM memory.

3. Terminal Status. There are two types of terminal status: primary and secondary.

    Primary terminal status returns seven bytes which report the status of some of the latching keys, various error and pending flags, and the following Terminal Configuration menu fields:

XmitFnctn(A)

SPOW(B)

InhEolWrp(C)

Line/Page(D)

InhHndshk(G)

InhDC2(H)

Secondary terminal status returns seven bytes which report the status of the following:

a. Memory Lock mode.

b. Buffer memory.

c. I/O firmware.

4. Device Status. This request returns three bytes that report the status of the external device.

5. Cursor Position Sensing. This request returns an escape sequence containing the row and column in which the cursor is located. Cursor position sensing is described in Section 4.

6. Command Completion Status. This request returns one character (S, F, or U) to indicate the completion status of the last command sent to the terminal. It can indicate satisfactory completion (S), failure (F), or interruption of the operation by the terminal operator pressing the [Return] key (U). Refer to Section 7 for detailed information.

7. Graphics Status. This request returns information on the selected one of 14 selectable graphics parameters.

8. Compatibility Mode Status. The terminal responds to three status requests in Compatibility Mode.

The escape sequence used for each of the above requests and the format of the returned status information is presented in the following paragraphs.

Status Transfer

All status requests are treated as block transfers. In response to a status request, the terminal transmits an escape sequence, followed by a series of data bytes, followed by a terminator.

Handshaking

The type of handshaking used is determined by the setting of the InhHndShk and InhDC2 fields of the Terminal Configuration menu as follows:

InhHndShk(G)	=	NO	DC1
Inh DC2(H)	=	YES or NO	
InhHndShk(G)	=	YES	No handshake
Inh DC2(H)	=	YES	
InhHndShk(G)	=	YES	DC1/DC2/DC1
Inh DC2(H)	=	NO	

Note that a status request escape sequence resets the "block trigger received" flag. This means, for example, that if you are using the DC1 handshake and the terminal receives a <DC1> followed by the request, it "forgets" that a block trigger was just received and thus will not send the data immediately. The terminal must receive another <DC1> before it will start the data transfer.

Terminators

The terminator character(s) following the status data is as follows:

Character Mode:	<CR> or <CR><LF>
Block Line Mode:	<CR> or <CR><LF>
Block Page Mode:	<Block Terminator>

In either Character mode or Block Line mode, the <CR><LF> is used if Auto Linefeed mode is enabled. In Block Page mode, the block terminator is as selected on the Terminal Configuration menu. The default block terminator is <RS>.

Status Transfer Priority

When handshaking is in effect and more than one status request is received, status data transfers are constructed and sent in the order of the block transfer priorities shown below. Only one status transfer occurs for each complete handshake, although more than one may be pending.

	PRIORITY OF BLOCK TRANSFERS
highest	Primary status (Ec ^)
.	Secondary status (Ec ~)
.	Device status (Ec &p<n>^)
.	Cursor sense (Ec ' or Ec a)
.	Transmit user keys (f1-f8) type
.	Display transfer (Enter key or Ec d)
.	Command completion status (S, F, or U returned)
.	Graphics status
lowest	Terminal ID and capabilities (Ec *s^)

If more than one status request of the same type is received, only the most recently received is acknowledged and sent when the handshaking is completed.

Interpreting Status

For status requests, the terminal returns an escape sequence followed by a string of bytes. The status information is contained in the lower bits of each byte. The upper bits are set so that the byte translates into an ASCII printing character (characters with ASCII decimal values from 32 to 126).

As an example, the format for primary, secondary, and device status requests is shown below. Notice that the upper four bits of each byte are set to "0011", which limits the ASCII decimal values to a minimum value of 48 (for these requests), well within the range of ASCII printing characters. The format for terminal capabilities status bytes is similiar.

ASCII Status Characters

ASCII Character	Binary
0	0011 0000
1	0011 0001
2	0011 0010
3	0011 0011
4	0011 0100
5	0011 0101
6	0011 0110
7	0011 0111
8	0011 1000
9	0011 1001
:	0011 1010
;	0011 1011
<	0011 1100
=	0011 1101
>	0011 1110
?	0011 1111

Terminal Identification

You request the terminal ID status by issuing the following escape sequence:

Ec *s ^ or Ec *s1 ^

The terminal responds by sending back the five-character string entered in the "Terminal Id" field of the Global Configuration menu. The default return string is "2390A".

Terminal Capabilities

Four requests can be issued for terminal capabilities: alphanumeric, graphics, interface capabilities, and amount of RAM memory. These requests are generated with the following escape sequence:

```
Ec *s <x>^
```

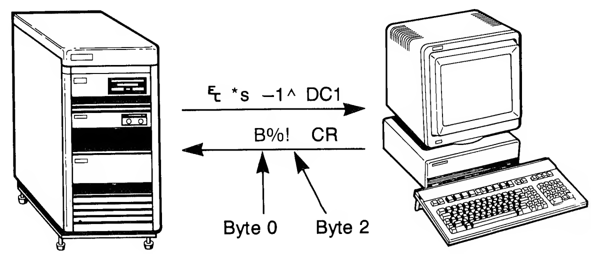
where "x" selects the request type, as follows:

X	REQUESTED INFORMATION
-1	Alphanumeric capabilities.
-2	Graphics capabilities.
-3	Amount of RAM memory available for downloading.
-4	Interface capabilities.
-5	HP-HIL interface capabilities.

The terminal responds with a string of bytes. The first byte indicates the number of status bytes in the response (this byte does not include itself in the count). The following byte(s) contain the requested data (figures 9-1 through 9-5).

If the "x" parameter is less than -5 (-6, -7, etc.), a single byte, the "@" character (01000000), is returned, indicating 0 status bytes. If "x" is greater than 1, then graphics status information is returned.

Figure 9-1. Terminal Capabilities (Alphanumeric-Typical) Status Example



BYTE	ASCII	BINARY	STATUS
0	B	01 000010	String length (no. of bytes) = 2
1	%	001 00101	Simple edit checks incorporated
			Extended edit checks not incorporated
			Modified data tags incorporated
			Forms cache not incorporated
			(Not used)
2	!	001 00001	Security enchancement supported
			{ 1 = Color incorporated 0 = Color not incorporated
			(Not used)

Figure 9-2. Terminal Alphanumeric Capabilities Status Bytes

BYTE 0 STRING LENGTH

8	7	6	5	4	3	2	1
0	1	0	0	0	0	1	0

Bits 6–1: String Length

BYTE 1 FORMS CHARACTERISTICS

8	7	6	5	4	3	2	1
0	0	1	0	0	1	0	1

Bit 5: Not Used

Bit 4: Forms Cache
1 = yes
0 = no

Bit 3: Modified Data Tags
1 = yes
0 = no

Bit 2: Extended Edit Checks
1 = yes
0 = no

Bit 1: Simple Edit Checks
1 = yes
0 = no

BYTE 2 DISPLAY CHARACTERISTICS

8	7	6	5	4	3	2	1
0	0	1	0	0	0	1/0	1/0

Bits 5–3: Not Used

Bit 2: Color
1 = yes
0 = no

Bit 1: Security Enhancement Supported
1 = yes
0 = no

Figure 9-3. Terminal Graphics Capabilities Status Bytes

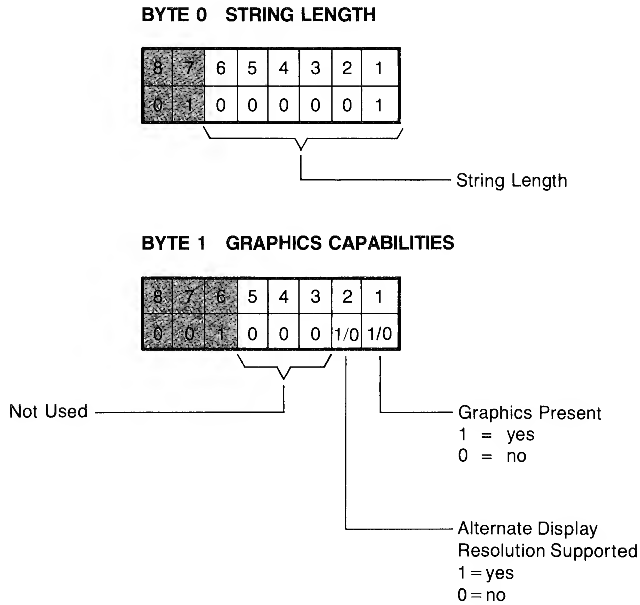


Figure 9-4. Memory Available for Downloading Code Status Bytes

BYTE 0 STRING LENGTH

8	7	6	5	4	3	2	1
0	1	0	0	0	0	1	0

String Length

BYTE 1 LEAST SIGNIFICANT BYTE

8	7	6	5	4	3	2	1
0	0	1	1/0	1/0	1/0	1/0	1/0

2^4 2^0

2^3 2^1

2^2

BYTE 2 MOST SIGNIFICANT BYTE

8	7	6	5	4	3	2	1
0	0	1	1/0	1/0	1/0	1/0	1/0

2^9 2^5

2^8 2^6

2^7

Note: To determine the memory size in K bytes, convert the binary number represented by bytes 1 and 2 to a decimal number, then multiply by 8 K.

Figure 9-5. Terminal Interface Capabilities Status Bytes

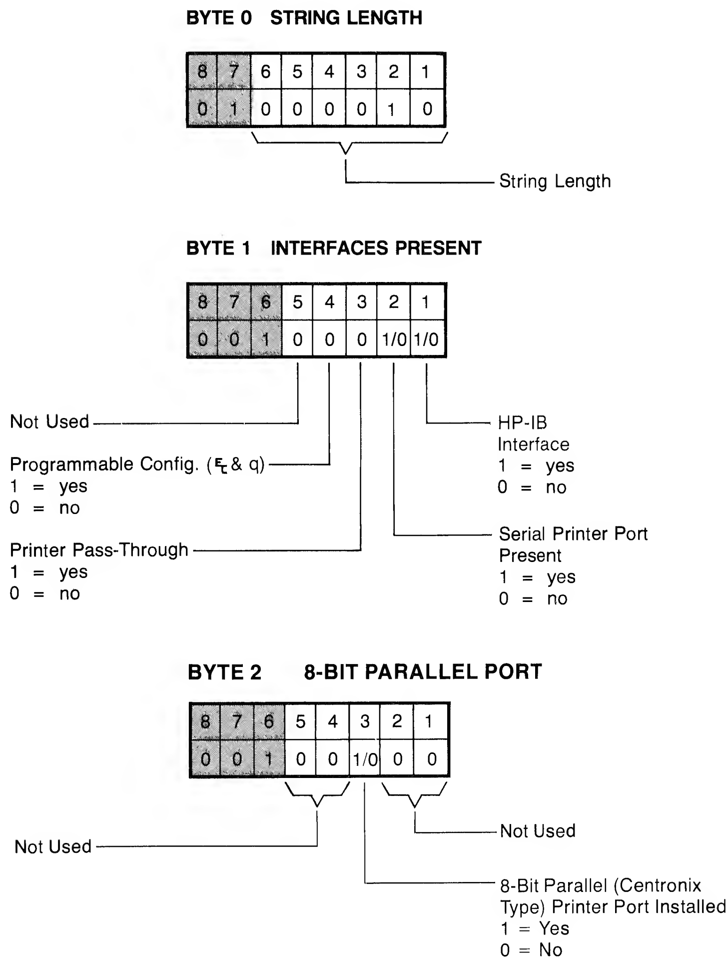
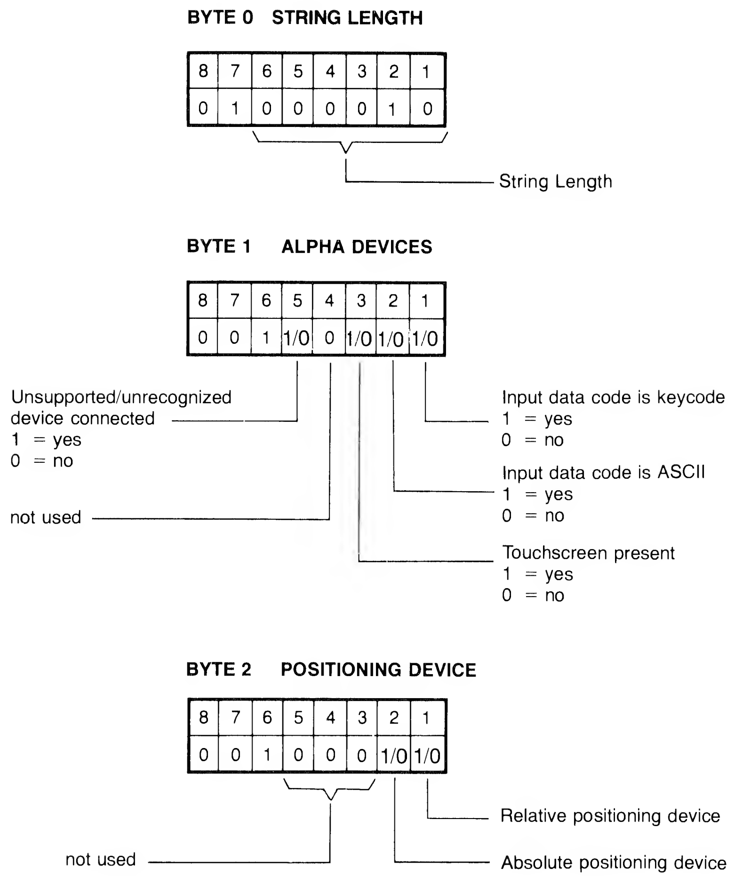


Figure 9-6. HP-HIL Capabilities Status Bytes



Terminal Status

Terminal status is made up of 14 status bytes (bytes 0–13) containing information such as display memory size, strap settings, configuration menu settings, and terminal errors. These 14 status bytes are displayed below the self-test screen pattern when the **TERMINAL TEST** key (in the "service keys" set of function keys) is pressed. There are two terminal status requests: primary and secondary. Each returns a set of 7 status bytes.

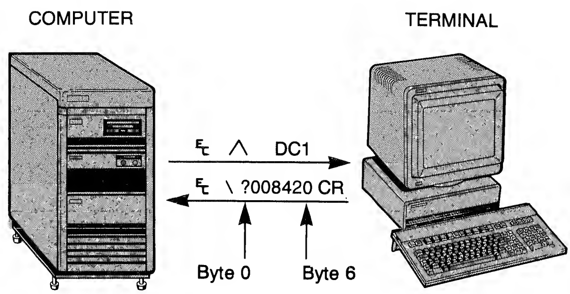
Primary Terminal Status

You request the first set of terminal status bytes (bytes 0–6) by issuing the following escape sequence:

```
Ec  ^
```

The terminal responds with an `Ec \`, and seven status bytes followed by a terminator. A typical primary terminal status request and response is illustrated in figure 9-7. The example assumes that the `DC1` handshake is being used and that the appropriate terminator is a `<CR>`. Figure 9-8 illustrates the function of each bit in each byte.

Figure 9-7. Terminal Primary Status Example.



BYTE	ASCII	BINARY	STATUS	
0	?	0011 1111	$\geq$15K bytes of display memory	
1	0	0011 0000	Function key transmission disabled Space overwrite latch disabled Cursor wrap around disabled Strapped for line	Configuration Straps A-H
2	0	0011 0000	Disable handshake Inhibit DC2 disabled	
3	8	0011 1000	Upper case only disabled Block mode disabled Auto line feed disabled Terminal sends secondary status (always 1)	
4	4	0011 0100	No cursor sense pending No function key pressed ENTER key has been pressed No secondary status request pending	
5	2	0011 0010	No data comm errors Last Self Test ok No loader checksum errors (always 0) No device errors	
6	0	0011 0000	No device status or device operation status pending	

Figure 9-8. Terminal Primary Status Bytes

BYTE 0 DISPLAY MEMORY SIZE

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

- Bit 1: 1K bytes
- Bit 2: 2K bytes
- Bit 3: 4K bytes
- Bit 4: 8K bytes

This byte specifies the amount of display memory available in the terminal.
Note that "1111" specifies 15K or more bytes.

BYTE 2 CONFIGURATION STRAPS E-H

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	0	0

- Bit 1: Strap E
- Bit 2: Strap F
- Bit 3: Strap G (DC2 Handshake)
  1 = yes (Enabled)
  0 = no (Disabled)
- Bit 4: Strap H (Inhibit DC2)
  1 = yes (Enabled)
  0 = no (Disabled)

BYTE 1 CONFIGURATION STRAPS A-D

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

- Bit 1: Strap A (Function Key Transmission)
  1 = yes (Transmitted)
  0 = no (Not transmitted)
- Bit 2: Strap B (Space Overwrite Latch)
  1 = yes (Enabled)
  0 = no (Disabled)
- Bit 3: Strap C (Inhibit End-of-Line Wraparound)
  1 = yes (Enabled)
  0 = no (Disabled)
- Bit 4: Strap D Page/Line
  1 = Page
  0 = Line

BYTE 3 LATCHING KEYS

8	7	6	5	4	3	2	1
0	0	1	1	1	1/0	1/0	1/0

- Bit 1: CAPS LOCK key
  1 upper case only
  0 upper and lower case
- Bit 2: BLOCK MODE Key
  1 block mode
  0 character mode
- Bit 3: AUTO LF Key
  1 = auto LF
  0 = no auto LF
- Bit 4: Terminal sends secondary status

Figure 9-8. Terminal Primary Status Bytes

BYTE 4 TRANSFER PENDING FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Cursor Sense Pending
1 = yes
0 = no

Function Key Pending
1 = yes
1 = no

ENTER Key Pending
1 = yes
0 = no

Secondary Status Pending
1 = yes
0 = no

BYTE 6 DEVICE TRANSFER PENDING FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	0	0	1/0	1/0

Device Status Pending
1 = yes
0 = no

Device Operation Status Pending
1 = yes
0 = no

BYTE 5 ERROR FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	1/0	0	1/0	1/0

Data Comm
1 = parity or buffer overflow error
0 = no error

Self Test
1 = no error
0 = error

Device Error
1 = error
0 = no error

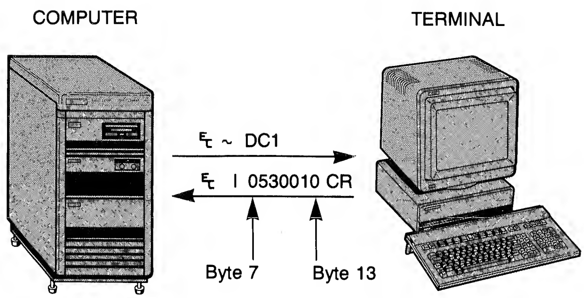
Secondary Terminal Status

You request the second set of terminal status bytes (bytes 7–13) by issuing the following escape sequence:

```
Ec ~
```

The terminal responds with an Ec |, and seven status bytes followed by a terminator. A typical secondary terminal status request and response is illustrated in figure 9-9. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a <CR>. Figure 9-10 illustrates the function of each bit in each byte.

Figure 9-9. Terminal Secondary Status Example.



BYTE	ASCII	BINARY	STATUS	
7	0	0011 0000	No buffer memory installed.	
8	5	0011 0101	(Always five)	
9	3	0011 0011	Auto terminate enabled Clear termintor enabled Self Test enabled	Configuration Straps J-L
10	0	0011 0000	Do not send escape codes to printer Compatibility mode off	
11	0	0011 0000	(Always zero)	
12	1	0011 0001	Data comm self-test inhibited	
13	0	0011 0000	Memory not locked	

Figure 9-10. Terminal Secondary Status Bytes

BYTE 7 BUFFER MEMORY

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- Bit 1: 1K bytes
- Bit 2: 2K bytes
- Bit 3: 4K bytes
- Bit 4: 8K bytes

Memory installed in addition to display memory that is available for use as data buffers (Always 0).

BYTE 8 TERMINAL FIRMWARE CONFIGURATION (always five)

8	7	6	5	4	3	2	1
0	0	1	1	0	1	0	1

- Bit 1: 1 = I/O firmware installed
- Bit 2: 0 = No APL Firmware
- Bit 3: 1 = Terminal identifies self
- Bit 4: 0 = Non-programmable terminal

BYTE 9 CONFIGURATION STRAPS J-M

8	7	6	5	4	3	2	1
0	0	1	1	0	1/0	1/0	1/0

- Bit 1: Strap J (Auto Terminate
  - 1 = yes (Enabled)
  - 0 = no (Disabled)
- Bit 2: Strap K (Clear Terminator)
  - 1 = yes (Enabled)
  - 0 = no (Disabled)
- Bit 3: Strap L (Self Test Inhibit)
  - 1 = yes (Inhibit test)
  - 0 = no (Allow test)
- Bit 4: Strap M

Strap M does not apply to the terminal.

BYTE 10 CONFIGURATION STRAPS N-R

8	7	6	5	4	3	2	1
0	0	1	1	0	1/0	1/0	1/0

- Bit 1: Strap N (Printer Escape Code Transfer)
  - 1 = open (Send ESC code)
  - 0 = closed (Do not send code)
- Bit 2: Strap P Compatibility Mode (Scaled)
  - 1 = Enabled
  - 0 = Disabled
- Bit 3: Strap Q Compatibility Mode (Unscaled)
  - 1 = Enabled
  - 0 = Disabled
- Bit 4: Strap R

Q	P	
0	0	Compatibility Mode off
0	1	Scaled Compatibility mode
1	0	Unscaled Compatibility mode
1	1	Compatibility Mode off

Figure 9-10. Terminal Secondary Status Bytes

BYTE 11   CONFIGURATION STRAPS S-V (always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- Bit 1: Strap S
- Bit 2: Strap T
- Bit 3: Strap U
- Bit 4: Strap V

Straps S-V do not apply to the terminal.

BYTE 13   MEMORY LOCK MODE

8	7	6	5	4	3	2	1
0	0	1	1	0	1/0	1/0	1/0

- Bit 1: 1 = locked in row ≠ 0; 0 = locked in row 0 (memory overflow protect)
- Bit 2: 1 = memory lock on; 0 = memory lock off
- Bit 3: 1 = memory full; 0 = memory not full

BYTE 12   CONFIGURATION STRAPS W-Z

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	1/0

- Bit 1: Strap W (Data Comm Test); 1 = yes (Inhibit); 0 = no (Allow)
- Bit 2: Strap X
- Bit 3: Strap Y
- Bit 4: Strap Z

Straps X, Y, and Z do not apply to the terminal.

Device Status

The status of the external device can be obtained by issuing a device status request. This request would typically be made following a print operation or after examining bytes 5 and 6 of the terminal status.

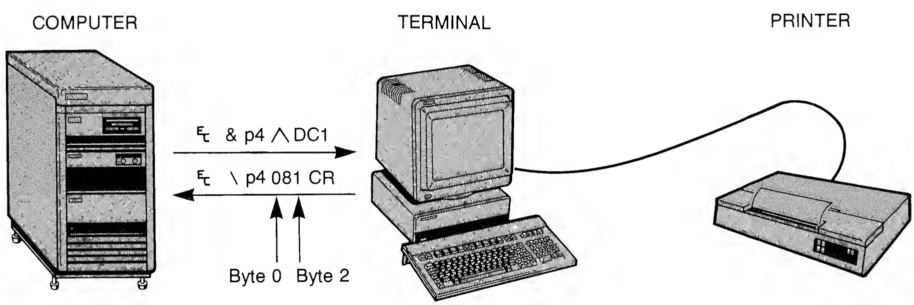
You request device status by issuing the following escape sequence:

```
Ec &p <device code> ^
```

where `<device code>` is either 4, 5, or 6. Any of these codes return the status of the external device, regardless of which interface module is installed in port 2.

The terminal responds with the sequence "`Ec\p <device code>`", followed by three status bytes, followed by a terminator. A typical device status request and response are illustrated in figure 9-11.

Figure 9-11. Device Status Example



BYTE	ASCII	BINARY	STATUS
0	0	0011 0000	No print error
1	8	0011 1000	Last print command performed
2	1	0011 0001	Printer is on-line

The device status bytes are shown in figure 9-12.

Figure 9-12. Device Status Bytes

DEVICE STATUS BYTES
(integral printer)

BYTE 0

8	7	6	5	4	3	2	1
0	0	1	1	0	0	1/0	1/0

Bit 1 — Status of Last Print Command
1 = last command failed
0 = last command successful

Bit 2 — Printer Error*
1 = yes
0 = no

*out of paper or printer gate open

BYTE 1

8	7	6	5	4	3	2	1
0	0	1	1	1/0	0	0	1/0

Bit 1 — Printer Busy
1 = device transfer in progress
0 = not busy

Bit 4 — Command Execution
1 = last command performed
0 = last command interrupted

BYTE 2

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	1/0

Bit 1 — Printer Connected
1 = yes
0 = no

Cursor Position Sensing

Refer to Section 4 for detailed information on cursor position sensing.

Command Completion Status

Refer to "Determining If Your Escape Sequence Command Has Been Successfully Performed" in Section 7 for information on command completion status.

Graphics Status

The status of 14 graphics parameters can be requested using an escape sequence. If an invalid escape sequence is used, the terminal will return the character string entered in the "Terminal Id" field of the Global Configuration menu (the default response is "2390A"). The escape sequence is as follows:

```
Ec*s <parm>^
```

where <parm> is as listed below:

PARM	READ REQUEST	RESPONSE
1	Device ID	Returns the string from the "Terminal Id" field of the Global Configuration menu.
2	Current pen position	X, Y, pen state
3	Graphics cursor position	X, Y
4	Graphics cursor position with wait	X, Y, key code
5	Display size	LLX, LLY, URX, URY, MMX, MMY
6	Device capabilities	b1, b2,.....b15, b16
7	Graphics text status	X size, Y size, origin, angle, slant
8	Zoom status	001, 0
9	Relocatable origin	X, Y
10	Reset status	Reset, b1,....b6, b7
11	Area shading capability	1, 8, 8
12	Dynamics capability	1,1
32	Graphics positioning device connected	0 or 1
33	Graphics cursor position	X, Y (Returned only if a tablet or mouse originates the position.)

The terminal responds to a status request with one or more bytes of status information, in ASCII form, separated by commas and followed by a terminator. Coordinates are returned in a fixed format, consisting of a sign and five digits (for example, +00100, -01234). This enables use of simple input statements without the need to mask or shift bits.

If the DC1 handshake protocol is enabled ("No" is entered in the "InhHndShk(G)" and "Yes" or "No" in the "Inh DC2(H)" fields of the Terminal Configuration menu — refer to Section 10), the status block is not actually sent until receipt of a DC1 character. If the DC1character is used, only one graphics status request can be enabled while the terminal is waiting for a DC1. When the DC1 is received, the last graphics status block requested will be sent.

While the terminal is waiting for the DC1, the Device Status Pending bit is set (byte 6 of the primary terminal status byte).

The terminal configuration determines the terminating character which is sent following the status block (CR, CR LF, or RS). Graphics status requests turn on an echo suppress mode in the terminal. This prevents information echoed back from the computer from being displayed on the screen. Once a graphics status block has been sent, characters received by the terminal will not be displayed until one of the following control characters is received: BELL, BS, CR, Ec, GS, HT, LF, RS, US, or VT. With the exception of CR and LFthe terminating control code itself will be executed.

The terminal expects the status information to be echoed and uses the terminating control character to turn off the suppress echo mode. If the computer doesn't echo the status back, a suitable control character must be returned to the terminal to turn off the suppress echo mode.

Read Device ID (Parameter=1)

When you request a device ID, the terminal responds with its Hewlett-Packard model number. The escape code is:

```
Ec*s 1^
```

The terminal returns the character string entered in the "Terminal Id" field of the Global Configuration menu, or "2390A" (default selection).

Read Current Pen Position (Parameter=2)

The pen position and status are returned as a string of ASCII characters when the following escape sequence is received:

```
Ec*s 2^
```

The terminal responds: "<X>, <Y>, <status>, <terminator>"

where:

<X>	=	X coordinate
<Y>	=	Y coordinate
<status>	=	0 = pen up 1 = pen down

For example, assume the pen is at 360, 80, the pen is up, and the terminal is set for the DC1 handshake with CR as the terminator:

The computer sends: "Ec*s2^<terminator>DC1"

The terminal responds: "+00360,+0080,0CR"

Read Graphics Cursor Position (Parameter=3)

The graphics cursor position is returned as a string of ASCII characters when the terminal receives the following escape sequence:

```
Ec*s 3^
```

The terminal responds: `<X>,<Y>CR`

where: `<X>` = X coordinate
`<Y>` = Y coordinate

With the cursor positioned in the lower left corner of the screen and `CR` as the terminator, the terminals response is:

```
+00000,+00000CR
```

Read Cursor Position with Wait (Parameter=4)

When the terminal receives this request, it waits until the user strikes a key before it returns the response. This allows the user to position the cursor before the cursor position is returned. The ASCII code for the key struck (not the actual character) is also returned. The code is returned as three digits. For example, striking an upper case A would return `065`. Only displayable character keys (keys with ASCII codes from 32 to 126) will produce a response.

If not already on, the graphics cursor will be turned on. If an escape sequence is received by the terminal after it has received the Read Cursor With Wait command and before a key is struck, the Read Cursor With Wait command will be aborted and the new sequence executed. The escape sequence follows:

```
Ec*s4^
```

The terminal responds: `<X>, <Y>, <key code> <terminator>`

where:	`<X>`	= X coordinate
	`<Y>`	= Y coordinate
	`<key code>`	= ASCII decimal value for the key struck

For example, if you position the cursor to the lower left corner of the screen, then press the [A] key, the terminal will respond:

```
+00000,+00000,065CR
```

Read Display Size (Parameter=5)

This request returns the coordinates of the lower left and upper right corners of the display area. It also returns the number of pixels per millimeter in the vertical and horizontal dimensions. The escape sequence is as follows:

```
Ec*s5^
```

The terminal responds: `<LLX>,<LLY>,<URX.,<URY>,<MMX>,<MMY><terminator>`

where: `<LLX>`, `<LLY>` = Lower left and upper right X coordinates

`<LLY>`, `<URY>` = Lower left and upper right Y coordinates

`<MMX>`, `<MMY>` = Number of pixels per millimeter in the horizontal and vertical dimensions (five digits and a decimal point)

The terminal will always return one of two fixed responses, depending on whether it is in low or high resolution mode. The lower left corner always has coordinates 0,0; the upper right corner has coordinates 511, 389 (low resolution) or 639, 399 (high resolution). There are approximately two pixels per millimeter.

Terminal Response (low resolution): `+00000,+00000,+00511,+00389,00002.,00002.<terminator>`

Terminal Response (high resolution): `+00000,+00000,+00639,+00399,00003.,00003.,<terminator>`

Read Device Capabilities (Parameter=6)

The device capabilities request returns a list of indications of the presence or absence in the terminal of a set of graphics-related features. If a feature is absent, the terminal returns a 0 for the feature. The escape sequence is as follows:

```
Ec*s6^
```

The terminal responds:

```
<b1>,<b2>,<b3>,<b4>,<b5>,<b6>,<b7>,<b8>,
<b9>,<b10>,<b11>,<b12>,<b13>,<b14>,
<b15>,<b16><terminator>
```

where: (normal selection marked with an asterisk (*)):

(* Normal selection for a monochrome terminal)
(# Normal selection for a color terminal)
($ Normal selection for both types of terminal)

<b1> = Clear Display
- 0 = no clear
- 1 = paper advance
- 2 = clear (total erase)
- $ 3 = partial clear by area

<b2> = Number of pens:
- * 1
- # 8

<b3> = Color Capability
- * 0 = black or white
- 1 = gray levels
- # 2 = color

<b4> = Color Level Capability (intensity levels for each primary color—RGB):
- * 2
- # 4

<b5> = Area Shading (refer to Read Area Shading Capability (Parameter=11))

0 = no

$ 1 = yes

<b6> = Not Used (0,0)

<b7>

<b8> Dynamic Modification (refer to Read Graphics Modification Capability (Parameter=12))

0 = no

$ 1 = yes

<b9> = Graphics Character Size

0 = fixed

$ 1 = integer multiples of the basic cell size

2 = any size

<b10> = Graphics Character Angles

0 = fixed

$ 1 = multiples of 90 degrees

2 = multiples of 45 degrees

3 = any angle

<b11> = Graphics Character Slant

0 = fixed

$ 1 = 27 degrees

2 = any angle

<b12> = Dot-Dash Line Patterns

0 = none

1 = predefined only

$ 2 = user-defined and predefined

<b13> = Not Used (0,0,0,0)
thru
<b16>

The terminal will always respond:

Monochrome:

```
3,1,0,0,1,0,0,1,1,1,1,2,0,0,0<terminator>
```

Color:

```
3,8,2,4,1,0,0,1,1,1,1,2,0,0,0<terminator>
```

Read Graphics Text Status (Parameter=7)

The terminal returns the current text size, orientation, slant, and type of justification. The escape code is as follows:

```
Ec*s7^
```

When it receives this escape code, the terminal returns:

```
<X size>,<Y size>,<origin>,<angle>,<slant>,
<terminator>
```

where:

<X size> = Horizonal dimension of the character cell, in pixels (sign plus 5 digits)

<Y size> = Vertical dimension of the character cell, in pixels (sign plus 5 digits)

<origin> = Position of text relative to cursor (refer to the text origin escape sequence, Ec*m <origin>q) (one digit)

<angle> = Text angle 0, 90, 180, or 270 (five digits and a decimal point)

<slant> = 00000. or 00027. degrees

Example of terminal response:

```
+00007,+00010,1,00090,00027.CR
```

Read Zoom Status (Parameter=8)

This request returns the zoom setting. Since the terminal doesn't have the zoom feature, it always returns constant values. The escape sequence is as follows:

```
Ec*s 8^
```

The terminals response is:

```
<zoom size>,<zoom on/off><terminator>
```

where:

`<zoom size>` = 001.
`<zoom on/off>` = 0 (for off)

Terminal response is always:

```
001.,0CR
```

Read Relocatable Origin (Parameter=9)

The position of the relocatable origin is returned as X and Y coordinates. The escape sequence is as follows:

```
Ec*s 9^
```

The terminals response is:

```
<X>,<Y><terminator>
```

With the origin set at the screens lower left corner, the terminal responds:

```
+00000,+00000CR
```

Read Reset Status (Parameter=10)

You can determine whether or not the terminal has undergone a hard reset or power on since the last time the reset status was checked. If so, you may want to reestablish the terminal settings and display content before resuming terminal operations. In addition to this information, the terminal returns an additional seven bytes, which are not used. The escape sequence is as follows:

```
Ec*s 10^
```

The terminal responds:

```
<reset>,<b1>,<b2>,<b3>,<b4>,<b5>,<b6>,<b7>
<terminator>
```

where:

<reset>	=	0	No hard reset or power on since last check
		1	Hard reset or power on has occurred since last check
<b1>-<b7>	=	(Not used)	

Read Area Shading Capability (Parameter=11)

The following escape sequence reads the terminals area filling capability. The terminals response to this request is fixed.

```
Ec*s 11^
```

The terminals response is always:

```
2,8,8<terminator>
```

where:

2 = Area filled can be a polygon

8 = Fill pattern is eight pixels wide

8 = Fill pattern is eight pixels high

Read Graphic Modification Capabilities (Parameter=12)

The following escape sequence reads a fixed response from the terminal, indicating the terminals ability to change selected portions of the display.

```
Ec*s 12^
```

The terminal will always respond:

```
1,1<terminator>
```

These two bytes indicate the terminal has selective erase and complement capabilities.

Tablet Identification (Parameter = 32)

The following escape sequence determines whether a graphics positioning device is connected to the terminal.

```
Ec*s 32^
```

A "1" is returned if a graphics positioning device is attached to the terminal. A "0" is returned if no such device is attached.

Read Graphics Cursor Position With Wait (Parameter = 33)

This request is the same as that for "parameter=4" except that, instead of waiting until a keyboard key is struck to return the cursor position, the terminal waits until the tablet stylus is clicked or until the button on the mouse is pressed. The escape sequence is as follows:

```
Ec*s 33^
```

Any Other Parameter

A request for any parameter which has not been assigned causes the terminal to respond with it's ID (the string entered in the "`Terminal Id`" field of the Global Configuration menu). This is to prevent an invalid status request from tying up the requesting computer while waiting for a response.

Compatibility Mode Status

In Compatibility mode, the terminal responds to three status requests:

- Read status and alphanumeric cursor position.
- Read graphics cursor position.
- Read graphics cursor position when a keyboard key is struck.

Read Status and Alphanumeric Cursor Position

The escape sequence with which a program requests status information and the alphanumeric cursor position is:

```
Ec EQ
```

The terminal returns the information in the following format:

```
<status byte><HI X><LO X><HI Y><LO Y>
 <terminator>
```

where:

`<status byte>` indicates:

- The ready status of a hardcopy unit connected to the terminal.
- Alphanumeric or Graphics mode.
- Whether the margin used is margin 1 or 2.

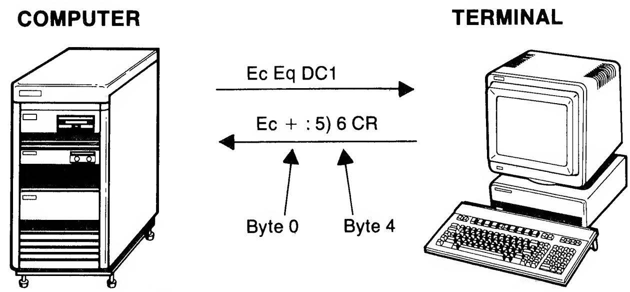
`<HI X>` `<LO X>` `<HI Y>` `<LO Y>` — four bytes which contain the X and Y coordinates of the alphanumeric cursor.

`<terminator>` one of seven ASCII control code terminators which indicate an action to be performed:

TERMI-NATOR	ASCII CODE	ACTION
GS	29	Enter Graphics mode.
US	31	Enter Alphanumeric mode.
BS	8	Backspace. Moves alphanumeric cursor one space left (14 units).
HT	9	Horizontal tab. Moves alphanumeric cursor one space right.
CR	13	Carriage return. Ends Graphics mode.
LF	10	Linefeed. Moves alphanumeric cursor one line down (22 units).
VT	11	Vertical tab. Moves alphanumeric cursor one line up. (22 units).

Figure 9-13 illustrates an example of use of this escape sequence, and figure 9-14 shows the function of each bit of the status and cursor position bytes.

Figure 9-13. Compatibility Mode Status and Alphanumeric Cursor Position Example



BYTE	ASCII	BINARY	STATUS
0	+	0010 1011	Always 1 Margin 2 — X coordinate of alphanumeric cursor position is 512 or more. Graphics mode Hard copy unit ready
1	:	0011 1010	X coordinate = 853
2	5	0011 0101	
3	)	0010 1001	Y coordinate = 310
4	6	0011 0110	

Figure 9-14. Compatibility Mode Status and Alphanumeric Cursor Position Bytes

BYTE 0 STATUS

8	7	6	5	4	3	2	1
0	0	1	0/1	0/1	0/1	0/1	1

- Bit 8: Depends on parity
- Bit 7: Always 0
- Bit 6: Always 1
- Bit 5: Hard copy unit
  0 = ready
  1 = not ready
- Bits 4–3: Mode
  10 = graphics
  01 = alpha
- Bit 2: Margin
  0 = margin 1
  1 = margin 2
- Bit 1: Auxiliary Device (not used)

BYTE 1 ALPHANUMERIC CURSOR POS. (HI X)

8	7	6	5	4	3	2	1
0	0	1	0/1	0/1	0/1	0/1	0/1

- Bit 5: 2^9
- Bit 4: 2^8
- Bit 3: 2^7
- Bit 2: 2^6
- Bit 1: 2^5

BYTE 2 ALPHANUMERIC CURSOR POS. (LO X)

8	7	6	5	4	3	2	1
0	1	0	0/1	0/1	0/1	0/1	0/1

- Bit 5: 2^4
- Bit 4: 2^3
- Bit 3: 2^2
- Bit 2: 2^1
- Bit 1: 2^0

BYTE 3 ALPHANUMERIC CURSOR POS. (HI Y)

8	7	6	5	4	3	2	1
0	0	1	0/1	0/1	0/1	0/1	0/1

- Bit 5: 2^9
- Bit 4: 2^8
- Bit 3: 2^7
- Bit 2: 2^6
- Bit 1: 2^5

BYTE 4 ALPHANUMERIC CURSOR POS. (LO Y)

8	7	6	5	4	3	2	1
0	0	1	0/1	0/1	0/1	0/1	0/1

- Bit 5: 2^4
- Bit 4: 2^3
- Bit 3: 2^2
- Bit 2: 2^1
- Bit 1: 2^0

Read Graphics Cursor Position

The escape sequence for reading the graphics cursor position is:

`Ec SB` followed by `Ec EQ`

where:

`SB` and `EQ` are the ASCII control codes with the decimal codes 26 and 5, respectively.

The terminal's response is in the following format:

`<HI X><LO X><HI Y><LO Y><terminator>`

where:

`<HI X>`
`<LO X>`
`<HI Y>`
`<LO Y>`
	the four bytes containing the X and Y coordinates of the graphics cursor position.

`<terminator>`	one of seven ASCII control codes indicating an action to be performed (refer to the explanation for `<terminator>` in the previous discussion of "Read Status and Alphanumeric Cursor Position").

Read Graphics Cursor Position When Key is Struck

This escape sequence enables a program to read the graphics cursor position when a keyboard key is struck. The escape sequence is:

`Ec SB`

where:

`SB` is the ASCII control code with the decimal code 26.

When it receives this escape code, the terminal returns the graphics cursor position in the following format:

```
<key><HI X><LO X><HI Y><LO Y><terminator>
```

where:

`<key>`	is the identity of the keyboard key struck to trigger the action.

`<HI X>`
`<LO X>`
`<HI Y>`
`<LO Y>`
	the four bytes containing the coordinates of the alphanumeric cursor position.

`<terminator>`	one of seven ASCII control codes which indicate an action to be performed (refer to the explanation of `<terminator>` in the previous discussion "Read Status and Alphanumeric Cursor Position").

Graphics Color Status

By sending a palette number, mix number, and request mask in an escape sequence, a program can request the following information from the terminal:

1. Has the indicated palette been defined?
2. Is the indicated palette the currently selected palette?
3. Which color selection method (RGB or HSL) is used on the indicated palette?
4. What is the number of the currently selected palette?
5. Return the RGB values for the indicated mix.
6. Return the HSL values for the indicated mix.

The escape sequence for requesting this information is as follows:

```
Ec*v <mask> <mix> <palette> ^
```

where:

`<mask>`: a decimal number in the range −32768 to 32767 which is converted to a binary number to select requests from the list which follows. Only the six least significant bits are used. If the number is 0, no information is returned.

The request associated with a bit in the binary number is selected with a "1", and not selected with a "0".

MASK BIT	MEANING	RESPONSE
0	Does the palette exist?	1 = yes 0 = no
1	Is the palette the currently loaded palette?	1 = yes 0 = no
2	Color selection method?	0 = RGB 1 = HSL
3	Number of currently loaded palette?	+nnnn (0–15)
4	Mix RGB values?	R = n.nnnn, G = n.nnnn, B = n.nnnn
5	Mix HSL values?	H = n.nnnn, S = n.nnnn, L = n.nnnn

The format in which the RGB or HSL data is returned is "n.nnnn,n.nnnn,n.nnnn".

When more than one mask bit is set to 1 (more than one request made), the responses are listed serially, separated by commas. The response to the least significant 1 bit in the mask is listed first, and the most significant last.

<mix> — the number of the mix for which information is requested when mask bit 4 or 5 is set. If a mix is not specified when bit 4 or 5 is set, information is returned for mix 7.

<palette> — the number of the palette for which information is requested when one or more of mask bits 0, 1, 4 or 5 is set. If any of these bits is set with no palette specified, information on the system palette (palette 0) is returned.

Note

If a palette number is specified, a mix number must also be included in the escape sequence. If a palette is specified but no mix is specified, the palette number will be interpreted as the mix number, resulting in erroneous information.

Example: This example lists a program, COLRSTAT, and its output, which consists of the terminal responses to mask decimal numbers 1–35 for mix 1, palette 0.

```
COLRSTAT
    10 DIM A$[80]
    15 PRINT " MASK NO."
    16 PRINT "(DECIMAL)","           RESPONSE(S)"
    17 PRINT
    20 FOR M=1 TO 35
    30   PRINT '27"*v";M;"1 0^"
    40   LINPUT A$
    50   PRINT "  ";M;"          ";A$
    60 NEXT M
    70 END
```

MASK NO. (DECIMAL)	RESPONSE(S)
1	1
2	1
3	1,1
4	0
5	1,0
6	1,0
7	1,1,0
8	+00000
9	1, +00000
10	1, +00000
11	1,1, +00000
12	0, +00000
13	1,0, +00000
14	1,0, +00000
15	1,1,0, +00000
16	1.0000,0.0000,0.0000
17	1,1.0000,0.0000,0.0000
18	1,1.0000,0.0000,0.0000
19	1,1,1.0000,0.0000,0.0000
20	0,1.0000,0.0000,0.0000
21	1,0,1.0000,0.0000,0.0000
22	1,0,1.0000,0.0000,0.0000
23	1,1,0,1.0000,0.0000,0.0000
24	+00000,1.0000,0.0000,0.0000
25	1, +00000,1.0000,0.0000,0.0000
26	1, +00000,1.0000,0.0000,0.0000
27	1,1, +00000,1.0000,0.0000,0.0000
28	0, +00000,1.0000,0.0000,0.0000
29	1,0, +00000,1.0000,0.0000,0.0000
30	1,0, +00000,1.0000,0.0000,0.0000
31	1,1,0, +00000,1.0000,0.0000,0.0000
32	1.0000,1.0000,1.0000
33	1,1.0000,1.0000,1.0000
34	1,1.0000,1.0000,1.0000
35	1,1,1.0000,1.0000,1.0000

10 Block Data Transfers

Introduction

This section describes block data transfers from the terminal to the host computer. The types of block transfers (all of which occur in Remote mode) are as follows:

- Initiated by the [Enter] key (including Line Modify and Modify All mode transfers).
- Initiated by the "`Ec d`" escape sequence.
- Transfer of the definition string of a "Transmit" type function key in Remote mode.
- Transfer of one or more user key definition string(s) when the [Enter] key is pressed, or the "`Ec d`" sequence is received, while the User Key menu is displayed.
- Status data transfer.

The manner in which data is transferred depends on the combination of terminal modes. The significant modes are Block, Character, Line, Page, Format, Non-Format, Autoterm, Non-Autoterm, Line Modify, and Modify All. These modes are defined by the menu selections `Line/Page(D)`, `InhHndShk(G)`, `Inh DC2(H)`, `AutoTerm(J)`, and `ClearTerm(K)`, as well as function keys BLOCK MODE, FORMAT MODE, LINE MODIFY, and MODIFY ALL.

Handshaking

The `InhHndShk(G)` and `Inh DC2(H)` selections on the Terminal Configuration menu determine, in general, the type of handshaking to be used when transferring blocks of data from the terminal to the host computer.

Handshake Types

There are three possible handshakes:

1. No handshake, also known as a type 1, in which the terminal simply sends the data block.
2. A DC1 Trigger handshake (type 2), in which the host computer must trigger the block transfer with a DC1 character.
3. A DC1/DC2/DC1 handshake (type 3), in which the host computer opens the transfer with a DC1 trigger the terminal signals a transfer ready with a DC2 as a warning, the host computer enables the transfer and signals that it is ready to receive a block with a second DC1 trigger, and finally the data block is transfered.

Handshake Type Selection

Along with the InhHndShk(G) and Inh DC2(H) straps, the Line/Page(D) selection on the Terminal Configuration menu and Character/Block mode function key selection determine the handshake type for some data transfers. Table 10-1 shows the configuration menu selections used to seleect a handshake type for a given mode. Table 10-2 indicates the current handshake type for any combination of strap settings.

Table 10.1 Handshake Selection by Configuration

H1 = no handshaking
H2 = DC1 handshaking
H3 = DC2 handshaking

TRANSFER TYPE	HANDSHAKE TYPE	MODE	G	H
ENTER key	H1	Character	No	—
	H1	—	—	Yes
	H2	(Option not available)		
	H3	Block	—	No
	H3	—	Yes	No
Status and Ec c	H1	—	Yes	Yes
	H2	—	No	—
	H3	—	Yes	No
User-defined function key ("T" type)	H1	Block Page	—	Yes
	H1	—	Yes	Yes
	H2	Block Line	No	—
	H2	Character	No	—
	H3	—	Yes	No
	H3	Block Page	—	No
Modify modes	H1	Character	—	Yes
	H1	Character	No	No
	H2	(Option not available)		
	H3	Character	Yes	No

Table 10-2. Handshake-Type Interpretation from Configuration

DEFINITIONS:

G = InhHndShk(G) strap
B = Block Mode (1)
Character mode (0)

H = Inh DC2(H) strap
D = Line (0)/Page (1) mode
(D) strap

H1 = no handshaking
H2 = DC1 handshaking
H3 = DC2 handshaking

CONDITIONS				TYPE OF TRANSMISSION														
				ENTER KEY			Ec d			STATUS KEY			FUNCTION KEY			MODIFY MODES		
G	H	B	D	H1	H2	H3	H1	H2	H3	H1	H2	H3	H1	H2	H3	H1	H2	H3
0	0	0	0	1	—	—	—	1	—	—	1	—	—	1	—	1	—	—
0	0	0	1	1	—	—	—	1	—	—	1	—	—	1	—	1	—	—
0	0	1	0	—	—	1	—	1	—	—	1	—	—	1	—	—	—	—
0	0	1	1	—	—	1	—	1	—	—	1	—	—	—	1	—	—	—
0	1	0	0	1	—	—	—	1	—	—	1	—	—	1	—	1	—	—
0	1	0	1	1	—	—	—	1	—	—	1	—	—	1	—	1	—	—
0	1	1	0	1	—	—	—	1	—	—	1	—	—	1	—	—	—	—
0	1	1	1	1	—	—	—	1	—	—	1	—	1	—	—	—	—	—
1	0	0	0	—	—	1	—	—	1	—	—	1	—	—	1	—	—	1
1	0	0	1	—	—	1	—	—	1	—	—	1	—	—	1	—	—	1
1	0	1	0	—	—	1	—	—	1	—	—	1	—	—	1	—	—	—
1	0	1	1	—	—	1	—	—	1	—	—	1	—	—	1	—	—	—
1	1	0	0	1	—	—	1	—	—	1	—	—	1	—	—	1	—	—
1	1	0	1	1	—	—	1	—	—	1	—	—	1	—	—	1	—	—
1	1	1	0	1	—	—	1	—	—	1	—	—	1	—	—	—	—	—
1	1	1	1	1	—	—	1	—	—	1	—	—	1	—	—	—	—	—

The values in table 10-2 reduce to the equations below (the symbol "~" indicates a logical NOT function):

ENTER key:	H1	= (~G)(~B) + H
	H2	= 0
	H3	= (~H)(G + B)
Status and Ec d:	H1	= GH
	H2	= ~G
	H3	= G(~H)
Transmit function key:	H1	= H(G + BD)
	H2	= ~G(~B + ~D) = ~(G+BD)
	H3	= ~H(G + BD)
Modify modes:	H1	= (~G + H)(~B)
	H2	= 0
	H3	= (~H)(G)(~B)

Data Transfer Priority

When handshaking other than "none" is in effect, and during receive mode for a half-duplex datacomm configuration, more than one status request or other form of block transfer may be pending while waiting on the completion of a handshake. In this situation, there is a prioritization of the responses rather than a first-in, first-out arrangement. The priority list is as follows:

- DC2 in response to a DC1 (as part of a DC1/DC2/DC1 handshake).
- Primary status.
- Secondary status.
- Device status.
- Cursor sense (absolute or relative).
- User-defined function key of "T" or "N" type.
- [Enter], [Return], or send display (Ec d).
- Command completion status code (S, F, or U).
- Terminal ID, terminal features, or graphics status.

The priority list is implemented by an internal queue. Only one request response is sent for each handshake completed. For device status, cursor sense, function key, and command completion code status transfers, there may be more than one request before the handshaking is done to allow the resulting transfer to take place.

DC1 Trigger Reset

Receipt of a status request or an "Ec d" escape sequence resets the terminals "DC1 Received" flag. As a result, when handshaking is selected, a DC1 must be received before the response is sent to the host computer, even if a DC1 was sent before the request or escape sequence.

Ending Characters Following DC2

When the DC1/DC2/DC1 handshake is used, the DC2 character may be followed by "ending characters". These characters are determined by the Line/Page(D) selection, on the Terminal Configuration menu. When the Line/Page(D) field is set to "Line", the ending is CR (LF); when set to "Page", there are no ending characters sent with the DC2.

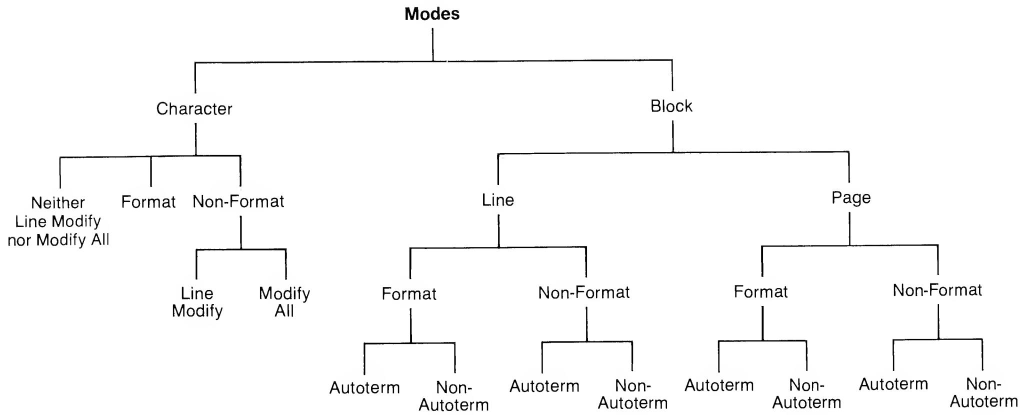
Enter Key Data Transfers

Data can be sent to the computer in multiple-character blocks by using the [Enter] key. The [Return] key can be used as a substitute for the [Enter] key by entering "Yes" in the RETURN=ENTER field of the Terminal Configuration menu.

When the terminal is in Remote mode, pressing the [Enter] key sets pending a block transfer of data from the workspace to the host computer. The keyboard is locked until the data transfer is completed.

The data transmitted depends on the combination of terminal modes. The modes significant to [Enter] key data transfer are shown in figure 10-1.

Figure 10-1. Modes Affecting ENTER Key Operation



In studying the following paragraphs, you should keep the following facts in mind:

- The data transfer is always terminated if a block terminator or a non-displaying terminator is encountered.
- If the data transfer is terminated by encountering a non-displaying terminator, that terminator may or may not be cleared, depending upon the setting of the `ClearTerm` field in the Terminal Configuration menu, as follows:

  `ClearTerm(K)` = `NO`: Do not clear the terminator.

  `ClearTerm(K)` = `YES`: Clear the terminator.

- Both the field separator and the block terminator are ASCII control codes and are selectable on the Terminal Configuration menu.
- When not in Format mode, you can insert a non-displaying terminator at the current cursor position by issuing an "`Ec _`" sequence. This escape sequence can be issued either through the keyboard or from a program executing in a host computer.
- In Format mode, non-displaying terminators may be inserted only into unprotected fields.

The significant factors involved in a data transfer are as follows:

- Data block start point.
- Data block end point.
- Type of data transferred.
- End character(s).
- Cursor location after the transfer is completed.

These factors are defined by the curret combination of terminal modes, which are selected using the function keys and by the entries in the Terminal Configuration menu. The applicable modes are illustrated in figure 10-1.

Data Transfer Event Sequence

Start Point Selection. The start point of the data block is determined by the terminal mode combination and the current cursor position. In many situations, the cursor is repositioned before the transfer is started; in others, the current cursor position becomes the start point.

The cursor might be repositioned either backward or forward. For forward repositioning, the start point is usually the start of the next unprotected or transmit-only field. When it is repositioned backward, a search is made for one of the following: column 1 of the current line, the first block terminator or non-displaying terminator, the home up position, the logical start-of-text, or the "`Start Col`" position selected on the Terminal Configuration menu. Then the cursor is located at this point, which becomes the start point.

When the `AutoTerm(J)` field is set to "`Yes`", the cursor is repositioned before handshaking is done (if handshaking is selected); in fact, before even the first `DC1` is received from the host. For all other menu field selections affecting cursor repositioning, the cursor is positioned after handshaking is completed.

Cursor positioning after an [Enter] keystroke is different from that of the send display (`Ec d`) escape sequence in that when the "`Ec d`" escape sequence is used, the cursor is never repositioned.

End Point Selection. The end point is determined by one or more "terminating agents", as listed below:

- Block terminator (BT).
- Non-displaying terminator (NDT).
- End of the line.
- End of the field (unprotected or transmit-only).
- End of the workspace.

In many cases, the data block can be terminated by more than one agent. In such cases, the first agent encountered ends the block.

Data Transfer. The type of data transferred depends on the current combination of terminal modes. Data types that might be transferred are as follows:

- ASCII displayable characters.
- ASCII control codes (characters).
- Video enhancement escape sequences.
- Alternate character set escape sequences.
- Field definition escape sequences.
- Edit check escape sequences.

Field definition types include: unprotected fields, protected fields, transmit-only fields, and security fields. Edit check types include: numeric-only, alphabetic-only, and all characters.

End Character Transfer. When the last character of the data block has been sent, the terminal adds one or more "end characters" to the data block. These are as follows:

- Block terminator.
- Carriage return.
- Line feed (if enabled).

The type of end character(s) transmitted depends on the terminal mode combination.

Cursor Location. The location of the cursor after the block transfer is completed depends on the terminal mode combination. It will be one of the following:

- Immediately following the end character.
- Column 1, with or without a line feed.
- Start column, with or without a line feed (modify modes data transfer).
- First column after the end of a field (when a field or portion of a field is transmitted).
- Last character position in the field transmitted.

Non-Format Mode Data Transfer

The following paragraphs contain some general rules applicable to data transfer when not in Format mode.

Data Transferred. Data transmitted in non-Format mode is as follows:

- ASCII displayable characters.
- ASCII control codes.
- Video enhancement escape sequences.
- Alternate character set escape sequences.
- Field definition escape sequences (including edit check escape sequences).

Video enhancements, alternate character set selections, field definitions, and edit checks in display memory are converted to escape sequences and sent with the ASCII characters.

Duplicate escape sequences are stripped from the data block before transmission. The duplications remain in the terminal display memory.

Cursor Start Point. The left margin setting has no effect in cursor positioning before or after a block transfer. For block transfers, the "home up" position for cursor repositioning is not the same as the "home up" for the escape sequence or keystroke. The block transfer "home up" is the top of the workspace and is not affected by the left margin setting as is the normal "home up".

When in Block mode and with the menu `AutoTerm(J)` field set to "`Yes`", the search backward, for a character (block terminator or non-displaying terminator) at which to locate the cursor start point, ends at the top of the workspace. The left margin setting is ignored.

If there is a memory-locked area and the cursor is outside of it when the [Enter] key is struck, the search backward for a cursor start point skips over the memory-locked area.

If the cursor is within a memory-locked area, a non-displaying terminator (NDT) is entered at the cursor location and the backward search includes the memory-locked area as well as any area preceding it.

If the top row of the workspace is below the memory-locked area and no block terminator or NDT is found in the memory-locked area, the cursor is set to the top of the workspace. Note that in a case where the cursor is in the memory-locked area to begin with, the inserted NDT would not be encountered and the transfer would terminate elsewhere.

Format Mode Data Transfer

The following paragraphs contain detailed information on the data fields recognized by Format mode, checks made on the fields before data transmission, the data block start point, and the type of data transferred.

Fields. The start point for a field is defined by a "start field" indicator. A "stop field" indicator defines the end of the field. A single field may extend beyond one line or row; such a field is called a continued field. For a continued field, a start field indicator is used in the first column of the second and each consecutive line occupied by the field. A stop field indicator is used at the end of the field.

If a start field is not in column 0 of a line following one containing a field with no stop field, the end of that line is the implicit end of field. No explicit stop field is needed.

The field separator or block terminator, as selected on the Terminal Configuration menu, is transmitted after the contents of the field, as would occur normally.

Field Checks. Two field attributes, the "required" attribute and the "modified data tag", are checked after the [Enter] key is pressed or at transmission time. Other attributes are checked when the fields are being filled by the user or when the field is to be exited.

Before the [Enter] keystroke is acknowledged, a check is done to ensure that all "required" fields have data entered in them. This check is done before any cursor repositioning and before any handshaking is done. They must all be in this state, even though none would be transmitted, given the current mode conditions and cursor position.

The "required" field check is not done for the "send display" (`Ec d`) data transfer.

Clearing of Terminators. Non-displaying terminators are cleared when tabbed over or when a "clear display" operation is executed. They are not cleared, however, when a cursor-move keystroke or cursor-movement escape sequence is executed.

Block terminators, like normal characters, are not cleared by tabbing.

Start Point. When in Block mode with the `AutoTerm(J)` strap set, the search backward for a block terminator or non-displaying terminator includes the protected fields. If a terminator is found in a protected field, the block transfer begins at the start of the next unprotected field. This is not a likely event, however.

Unlike the non-Format mode case, a memory locked area is searched (after the workspace is adjusted completely) and is treated as the rest of the workspace is. This is the case also for "home up"; the memory-locked area is only special in Format mode, in that it does not roll up or down.

Data Transmitted. In Format mode, only the ASCII characters (both control codes and displayable characters) in unprotected fields are transmitted.

When the terminal is configured to send only "modified" fields, data compression is performed. When in Character, Block Line, and Block Page modes, trailing blanks are eliminated from the field contents.

In Character and Block Line modes (but not Block Page mode), a field is sent to the host computer even though it has not been modified, that is, the MDT is not checked. In Block Page mode, only modified fields are sent.

When a field which has been modified is transmitted, the field separator character, selected on the Terminal Configuration menu, is sent for each unmodified field preceeding the modified one. If a field has been filled with blanks (for example, by "clear display" from the keyboard), then one blank is sent for each field; only trailing blanks are eliminated. A "clear display" from the host computer leaves the cleared fields unmodified.

ENTER Key Data Transfer Summary

Most of the preceding information is summarized in table 10-4.

Table 10-4. ENTER Key Block Transfer Summary

DEFINITIONS:

LF	=	Line Feed	FS	=	Field separator
CR	=	Carriage return	EOL	=	End of line
BT	=	Block terminator	EOF	=	End of field
NDT	=	Non-displaying terminator	EOW	=	End of workspace

Note:

1. Multiple entries in the "Terminating Agent" column indicate the data block will be terminated by whichever type of entry is encountered first.
2. An "LF", enclosed in parenthesis, indicates a line feed will occur, if enabled.
3. The entry "all ASCII chars" includes both displayable and control ASCII characters (decimal codes 0-127).

MODE	TRANSFER START POINT	TERMI-NATING AGENT	DATA TRANSFERRED	END CHAR	FINAL CURSOR LOCATION
CHARACTER NON-FORMAT (Neither Line Modify nor Modify All mode)	Column 1, current line.	BT NDT	All ASCII chars and the following Ec sequences: ■ Video enhancement ■ Alt char set ■ Field definition If no data to be transferred or cursor is at end of workspace when transfer is initated, only "end char" sent.	BT CR (LF)	Immediately following terminator.
		EOL	Same as above	CR (LF)	Column 1. LF, if enabled.
(Modify Line and Modify All Modes)	Logical start-of-text position, if one is defined. Else to "Start Col" as specified in Terminal Configuration menu.	BT NDT	All chars, including: ■ Control codes. ■ Video enhancement Ec sequences. ■ Alt char set Ec sequences. ■ Field definition Ec sequences. ■ Same as above.	BT CR (LF)	Immediately following terminator.
		EOL		CR (LF)	Column at which transmission began. LF, if enabled.

Table 10-4. ENTER Key Block Transfer Summary (continued)

MODE	TRANSFER START POINT	TERMI-NATING AGENT	DATA TRANSFERRED	END CHAR	FINAL CURSOR LOCATION
CHARACTER FORMAT	Current cursor position, if in an unprotected field, else start of next unprotected field.	EOF	All ASCII chars. If cursor is at end of workspace when transfer is initiated, only "end char" sent.	CR (LF)	First column after the end of the field sent.
		BT NDT	Same as above	BT CR (LF)	Immediately following terminator.
BLOCK LINE NON-FORMAT					
`AutoTerm (J) = NO Inh DC2 (H) = YES`	Column 1, current line.	BT NDT	All ASCII chars and the following `Ec` sequences; ■ Video enhancement. ■ Alt char set. ■ Field definition. If no data to be transferred or cursor is at end of workspace when transfer is initiated, only "end char" sent.	BT CR (LF)	Immediately following terminator.
		EOL	Same as above.	CR (LF)	Same as above.

Table 10-4. ENTER Key Block Transfer Summary (continued)

MODE	TRANSFER START POINT	TERMINATING AGENT	DATA TRANSFERRED	END CHAR	FINAL CURSOR LOCATION
`AutoTerm (J) = NO` `Inh DC2 (H) = NO`	Current cursor position	BT NDT	All ASCII chars and the following `Ec` sequences: ■ Video enhancement. ■ Alt char set. ■ Field definition. If no data to be transferred or cursor is at end of workspace when transfer is initiated, only "end char" sent.	BT CR (LF)	Same as above.
		EOL	Same as above.	CR (LF)	Same as above.
`AutoTerm (J) = YES`	NDT entered at current cursor position. Then cursor moved to first previous BT or NDT. If non found, cursor is homed (left margin ignored).	NDT	All ASCII chars and the following `Ec` sequences; ■ Video enhancement. ■ Alt char set. ■ Field definition. If no data to be transferred or cursor is at end of workspace when transfer is initiated, only "end char" sent.	BT CR (LF)	Same as above.
		EOL	Same as above.	CR (LF)	Column 1. LF, if enabled.
	If already an NDT at current cursor position, cursor remains at current position.	—	No data transmitted, only "end char".	BT CR (LF)	Cursor doesn't move.

Table 10-4. ENTER Key Block Transfer Summary (continued)

MODE	TRANSFER START POINT	TERMI-NATING AGENT	DATA TRANSFERRED	END CHAR	FINAL CURSOR LOCATION
BLOCK LINE FORMAT					
`AutoTerm (J) = NO`	Current cursor position if in an unprotected field, else start of next unprotected or transmit-only field.	BT NDT	All ASCII chars.	BT CR (LF)	Immediately following terminator.
		EOF	Same as above.	CR (LF)	First column following end of field.
`AutoTerm (J) = YES`	Cursor in protected field.	—	Bell sounds. No data transmitted.	BT CR (LF)	Cursor doesn't move.
	Cursor not currently in a protected field: An NDT entered at current cursor position. Then, cursor moved to first previous BT or NDT. If none, then cursor homed.	NDT	All ASCII chars. If no data to be sent, "end char" transmitted.	BT CR (LF)	Immediately following the terminator.
		EOF	Same as above.	CR (LF)	First column following end of field.
	If already an NDT at current cursor position,	—	No data transmitted, only "end char".	BT CR (LF)	Cursor doesn't move.

Table 10-4. ENTER Key Block Transfer Summary (continued)

MODE	TRANSFER START POINT	TERMINATING AGENT	DATA TRANSFERRED	END CHAR	FINAL CURSOR LOCATION
BLOCK PAGE NON-FORMAT					
AutoTerm (J) = NO Inh DC2 (H) = YES	Home up position (left margin ignored).	BT NDT EOW	All ASCII chars and the following Ec sequences; ■ Video enhancement. ■ Alt char set. ■ Field definition. Each line followed by CR LF (regardless of whether Auto Linefeed mode is enabled or disabled). Last line followed by "end char" after the usual CR LF. If no data to be sent, only "end char" is transmitted.	BT	Immediately following terminator.
AutoTerm (J) = NO Inh DC2 (H) = NO	Current cursor position	NDT EOW	Same as above.	BT	Same as above.

Table 10-4. ENTER Key Block Transfer Summary (continued)

MODE	TRANSFER START POINT	TERMINATING AGENT	DATA TRANSFERRED	END CHAR	FINAL CURSOR LOCATION
AutoTerm (J) = YES	An NDT entered at current cursor position. Then, cursor moved to first previous BT or NDT. Cursor homed, if none found (left margin ignored).	NDT	Same as above.	BT	Same as above.
	If already an NDT at current cursor position,	—	No data transmitted, only "end char".	BT CR (LF)	Cursor doesn't move.
BLOCK PAGE FORMAT					
AutoTerm (J) = NO Inh DC2 (H) = YES	Home up position.	BT NDT	All ASCII chars in unprotected and transmit-only fields. Each field transmitted as a separate block. Each field, except the last, is followed by a field separator. The last field is followed by the "end char".	BT	Immediately following terminator.
		EOW	If no unprotected or transmit-only fields found, only "end char" sent.	BT (LF)	Same as above.
AutoTerm (J) = NO Inh DC2 (H) = NO	Current cursor position.	BT NDT	Same as above.	BT	Same as above.
		EOW	If no unprotected or transmit-only fields found, only "end char" sent.		

Table 10-4. ENTER Key Block Transfer Summary (continued)

MODE	TRANSFER START POINT	TERMI-NATING AGENT	DATA TRANSFERRED	END CHAR	FINAL CURSOR LOCATION
AutoTerm (J) = YES	Cursor currently in an unprotected field: an NDT entered at current cursor position. Then, cursor moved to first previous BT or NDT. If none, then cursor homed.	NDT	Same as above.	BT	Same as above.
		EOW	Same as above.		
	Cursor currently in a protected field.	—	Bell sounds. No data transmitted.	BT	Cursor doesn't move.
	If already an NDT at current cursor position,	—	No data transmitted, only "end char".	BT CR (LF)	Cursor doesn't move.

Send Display (Ec d)

From a program executing in a host computer, you can trigger a block transfer of data from the workspace to the host computer by issuing the following escape sequence:

```
Ec d
```

This escape sequence is only responded to when received over a datacomm line; it is ignored if entered through the keyboard or issued from a user key (unless Block mode is enabled). With the following exceptions, the resultant data transfer is performed as though the [Enter] key had been pressed:

1. The cursor is not repositioned. The data transfer always begins at the current cursor position.
2. A non-displaying terminator is never inserted at the cursor position as part of the operation (the AutoTerm(J) configuration parameter is ignored).

The "Ec d" sequence also temporarily disables the keyboard so that the Enter key cannot be used until the current data transfer is completed. If the "Ec d" sequence is received while an Enter key data transfer is in progress, the escape sequence is ignored.

Note that an "Ec d" sequence resets the "block trigger received" flag. This means, for example, that if you are using the DC1 handshake and the terminal receives a <DC1> followed by the "Ec d", it "forgets" that a block trigger (the <DC1>) was just received and thus will not send the data immediately. The terminal must receive another <DC1> before it will start the data transfer.

User Key Definition String Transfer

When the User Key menu is displayed and the terminal is in Remote mode, both the Enter key and "Ec d" sequence cause one or more of the function key definitions to be transmitted, in the escape sequence form. The AutoTerm(J) and ClearTerm(K) entries on the Terminal Configuration menu are ignored under these circumstances.

The first key definition sent is the one on the line on which the cursor is located.

In Block Page mode, all of the function key definitions, from the cursor row to the end of the menu, are transmitted. Then, the cursor is positioned on the blank line following the last function key definition.

In Character or Block Line mode, only one key definition is transmitted, but the cursor is always moved to the start of

the next key definition, regardless of the state of Auto Linfeed mode. Block terminators in the definition string are treated the same as any other character. Non-displaying terminators are ignored.

Table 10-5 lists the end characters.

Table 10-5. End Characters for User Key Definition String Transfer

DEFINITIONS:

BT = Block terminator
CR = Carriage return
LF = Line feed
(LF) = Line feed, if enabled

MODE	EOL	EOB	EOW
Character or Block Line	CR (LF)	—	BT CR (LF)
Block Page (Format and Non-Format)	CR LF	CR LF BT	BT

Function Key Definition String Transfer

If a function key of Transmit type is pressed in Remote mode, its definition string will be transmitted to the host computer, as a block. It can also be transmitted using the following escape sequence:

```
Ec &f <x>E
```

where "x" is the identification number (1–8) of the function key.

When in Block Page mode, the end character is the configured block terminator; otherwise the ending is CR (LF).

Status Data Transfer

The status request types are as follows:

- Primary and secondary status.
- Device status.
- Cursor position sensing (both absolute and relative).
- Command completion status (S/F/U).
- Terminal identification.
- Terminal features.

These requests are all intiated by escape sequences, and they are all returned as fixed-length strings of ASCII characters.

The ending character is the configured block terminator when in Block Page mode, otherwise the ending is CR (LF).

Special Modes

Two modes, Auto Keyboard Lock and Send Cursor Position, are available for use under special circumstances. Auto Keyboard Lock mode can be used when the terminal is used with the X.25 protocol, and Send Cursor Position mode can be used for combinations of terminal modes which require that the cursor position be sent to the computer before the data transfer begins. Both modes are accessible only through escape sequences.

Auto Keyboard Lock Mode

Use of the X.25 protocol in a network in which the terminal is connected requires that the block of data sent is received and acted upon (in some way) before the terminal sends another data block. To ensure this, Auto Keyboard Lock

mode locks the terminal keyboard in a manner in which the keyboard will normally be unlocked only by the receiving host.

Auto Keyboard Lock mode is accessible only by the following escape sequences:

`Ec & k 1 K`	Enable Auto Keyboard Lock mode
`Ec & k 0 K`	Disable Auto Keyboard Lock mode

When disabled (the default mode), the terminal will act as currenty defined. When enabled, the user-definable function keys which have been assigned the Transmit Only attribute and the [Enter] key will leave it locked after the data transfer has taken place. The host computer can then unlock the keyboard (with the "`Ec b`" sequence) when it is ready to receive more data.

Note When the keyboard is unlocked, the keyboard input buffer is flushed, so that any data in it is lost.

A soft reset on the terminal will unlock the keyboard, but will not affect Auto Keyboard Lock mode.

A hard reset or power-on will put this mode in its default state (disabled).

The send display (`Ec d`) sequence is not affected by the Auto Keyboard Lock mode. However, the escape sequence which triggers the user-defined function keys (`Ec &f<n>E`) is affected by this mode, since it is defined to act as though the key selected in the escape sequence is depressed.

Send Cursor Position Mode

Under certain circumstances, when the [Enter] key is used, the current cursor position is required before the data is transferred. In Send Cursor Position mode, the current cursor position is supplied to the computer in the form

"Ec &a<xxx>c<yyy>R" inserted at the start of the data block. The cursor position (xxx, yyy) is in workspace-relative form (refer to Memory Addressing in Section 4), and is the same form as that for an absolute cursor sense status request (Ec a).

Send Cursor Position mode is valid only when the terminal is in Block mode.

The action of Send Cursor Position mode occurs when the [Enter] key (or the [Return] key when the Return=Enter field on the Terminal Configuration menu is set to YES) is pressed or when any user key of type Transmit Only is pressed.

The cursor position sent is that of the cursor before any repositioning is done, regardless of the entries on the Terminal Configuration menu. The block transfer obeys the usual handshaking conventions.

Send Cursor Position mode cannot be activated in a Send Display operation (initiated by the escape sequence "Ec d"). However, it can be activated for the function key triggering operation, initiated by the escape sequence "Ec &f <x>E".

A soft reset will leave Send Cursor Position mode unchanged. A hard reset will set it to its default (disabled) condition.

This mode is accessible through an escape sequence only. The sequences for enabling and disabling the mode are as follows:

```
Ec & x 1 C    Enable SCP mode

Ec & x 0 C    Disable SCP mode
```

These escape sequences cannot be combined with other "Ec &x" type sequences.

11 User-Definable Function Keys

Introduction

The keyboard contains 12 function keys; the first eight are labeled f1 through f8; the remaining four (f9 through f12) are the four unlabeled keys above the numeric keypad (figure 11-1). Each function key can be programmed with a character string (definition) of up to 80 characters (or 160 "mute" characters). Then, when the key is triggered, the character string is executed as though it was typed in from the keyboard. The definition can contain executable commands, such as Carriage Return, Line Feed, Backspace, etc., which will be executed when the key definition is executed.

The keys can be triggered as follows:

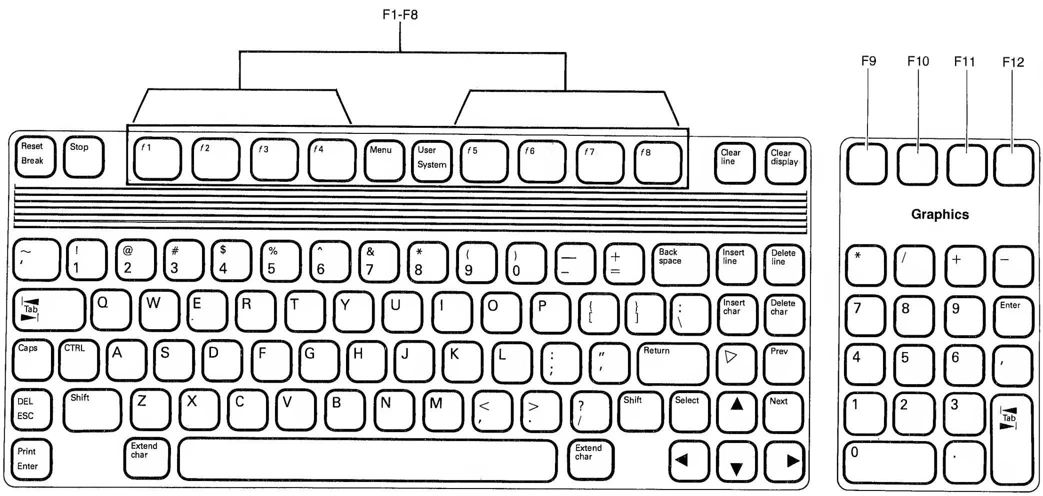
a. Pressing the key in User Key mode.

b. Pressing the key, together with the [Shift] key, in User Key mode.

c. From a program.

d. When a power on occurs (f9 through f12, only).

Figure 11-1. Function Keys Location



These keys are convenient to use when a character string must be entered often or repeatedly. Used this way, the keys are called "user keys". The keys can be controlled either locally, from the keyboard, or remotely, by a program executing on a host computer.

User Key Modes

Two modes are associated with the user keys: Definition mode and User Key mode.

Definition Mode

In Definition mode, the keys are assigned types, labels (f1–f8 only), and a definition. Also, keys f9–f12 can be selected for automatic execution when a power on occurs. The key type selects the disposition of the definition string: terminal, host computer, or as though entered from the keyboard. The labels assigned can be displayed at the bottom of the screen in User Key mode. The definition is executed when triggered, either by pressing the key in User Key mode or from a program.

User Key Mode

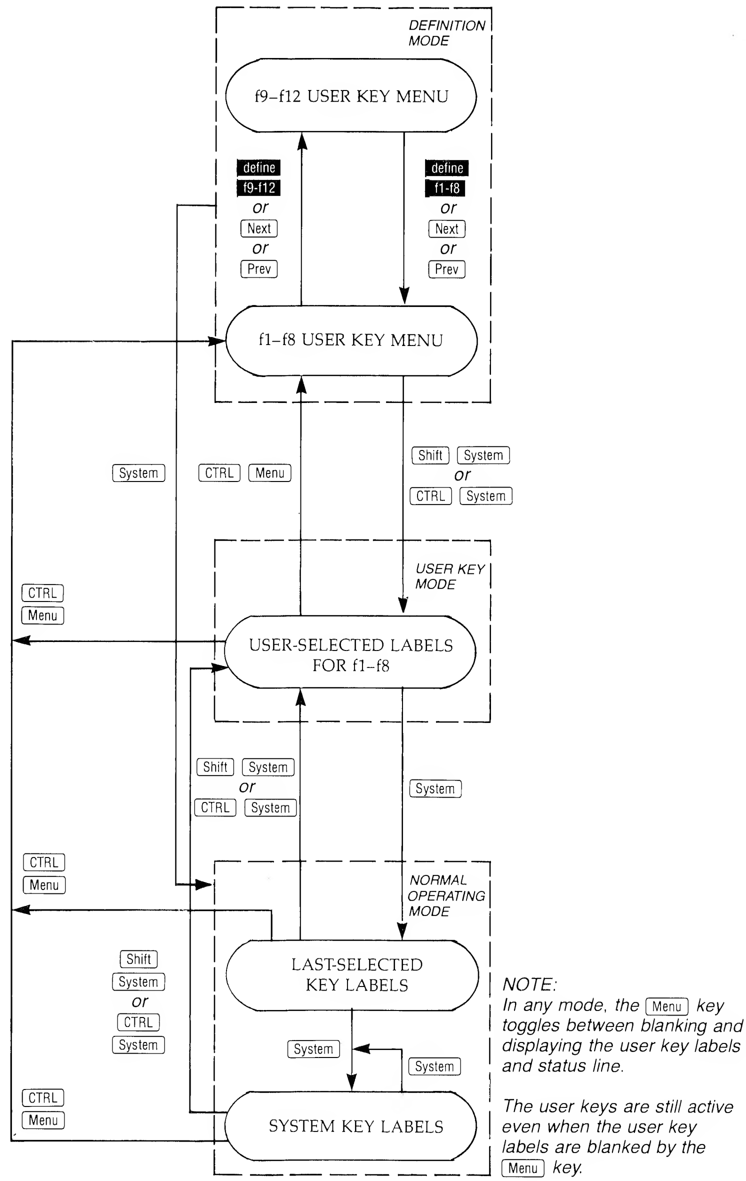
In User Key mode, the labels assigned to keys f1–f8 are displayed and the keys are active (ready for use). When the user key is pressed in User Key mode, the definition string is executed. The user key labels can be toggled on and off in User Key mode by pressing the [Menu] key. However, whether or not the labels are displayed, the keys remain active.

Use

The user keys can be used by entering Definition mode to define the user keys, then entering User Key mode, from normal operating mode, to use them. From normal operating mode, the default definition of a function key can be transmitted by pressing the key together with the [Shift] key.

In normal operating mode, the normal function key labels are displayed and active. User Key mode can be entered by pressing either the [Shift] or [CTRL] key simultaneously with the [System] key, and exited to normal operating mode by pressing the [System] key. See figure 11-2.

Figure 11-2. Access to Definition and User Key Modes



Defining the User Keys Locally (Definition Mode)

By "defined" it is meant that:

1. Each key has an operational attribute ('type') which may be one of three types:

Local:	executed locally at the terminal.
Transmit:	transmitted to a host computer.
Normal:	as though typed in from the keyboard.

2. Each of keys f1–f8 has an alphanumeric label which, in User Keys mode, is displayed with its companions across the bottom of the screen. These labels may be up to 16 displayable characters and may have one video enhancement selection at the beginning of each label half, allowing different enhancements for the top and bottom. Keys f9–f12 have no labels and are not represented by labels on the screen.

3. A string of ASCII alphanumeric characters and/or control codes may be assigned to each key. This definition string may contain explicit escape sequences (entered using Display Functions mode) but cannot contain any embedded or implicit ones (such as enhancements). It may also contain commands such as Carriage Return and Backspace. The maximum length is 80 characters, including escape sequences and commands, or 160 characters which use mute characters (refer to Appendix D for an explanation of mute characters). Any escape sequences or commands within the definition will take effect when the function key is triggered (executed).

   For type "N" keys, you can embed, in a user key definition, the escape sequence for entering or leaving Remote mode. Then all escape sequences in the definition string following the Remote mode sequence will be executed based on the new Remote/Local mode state.

4. Keys f9–f12 can be selected to be triggered, automatically, after a power on.

Entering Definition Mode

To initiate Definition mode and access the User Key Menu, press the CTRL and Menu keys simultaneously, or use the escape sequence "Ec j".

User Key Menus

The User Key menu for keys f1–f8 is shown, with the default values displayed, in figure 11-3. Figure 11-4 shows the menu for keys f9–f12.

Figure 11-3. User Key Menu for f1–f8, Showing Default Values

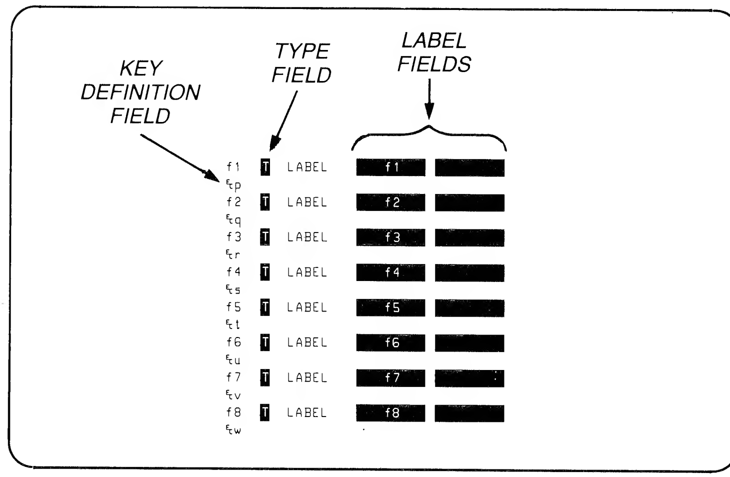
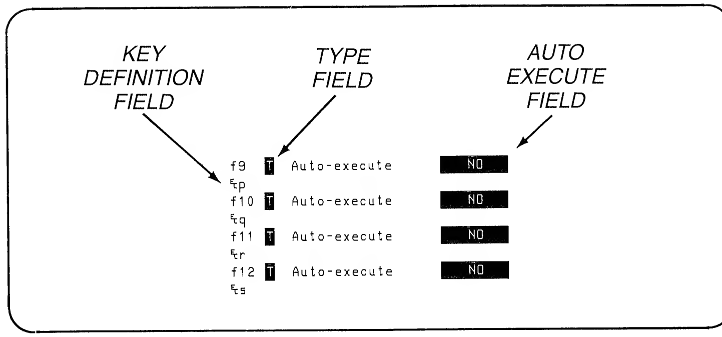


Figure 11-4. User Key Menu for f9–f12, Showing Default Values



Type Field. This one character field must contain only an uppercase L, T, or N, signifying Local, Transmit, and Normal.

a. A Local key is executed locally (at the terminal) only.

b. A Transmit key is transmitted to the host computer only.

c. A Normal key is executed in the same manner as data typed in from the keyboard. This means that if the terminal is in Local mode, the contents of the key are executed locally (displayable characters are sent to the display). If the terminal is in Remote mode and local echo is disabled (configuration item "Local Echo"), the contents of the key are transmitted to the host computer. If the terminal is in Remote mode and local echo is enabled (on), the contents of the key are both transmitted to the host computer and executed locally. If the terminal is in Block mode, the contents of the key are executed locally.

The type field cannot be changed using the alphanumeric keys; the NEXT CHOICE and PREVIOUS CHOICE function keys must be used.

Label Fields (f1–f8). There are two eight-character label fields for keys f1–f8. The first field supplies the upper half of the label displayed on the screen; the second field supplies the lower half. Each field can have a video enhancement, which is entered programmatically using an `Ec &f` escape sequence (described later in this section). When the User Key menu is displayed, it will not show any video enhancements.

Control codes and escape sequences may be entered into these fields only by using Display Functions mode.

Auto-Execute Field (f9–f12). Keys f9–f12 have a field for selection of automatic execution of the key definition at power on time. The options are YES and NO, selectable with the NEXT CHOICE and PREVIOUS CHOICE keys. These keys are executed in numerical order.

Key Definition Field. The entire line (80 characters) immediately below the type and label fields is available for specifying the character string that is to be displayed, executed, and/or transmitted whenever the key is either pressed or programmatically triggered. Control codes and escape sequences may be entered into this field only by using Display Functions mode.

Defining the Keys

One of two sets of function keys can be displayed in Definition mode, one for defining keys f1–f8 and a second set for defining keys f9–f12. The set displayed with the f1–f8 User Key menu is shown below.

f1	f2	f3	f4	f5	f6	f7	f8
	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES			DISPLAY FUNCTNS	define f9-f12

The set displayed with the f9–f12 User Key menu are shown below.

f1	f2	f3	f4	f5	f6	f7	f8
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	POWER ON VALUES		DISPLAY FUNCTNS	define f1-f8

The functions of the keys displayed with both User Key menus are listed in table 11-1.

Table 11-1. User Key Menu Function Keys

KEY	FUNCTION
NEXT CHOICE PREVIOUS CHOICE	The menu Type Field has a list of three choices: L, T, and N. These two keys can be used to cycle backward or forward through the list, enabling selection of the choice for the display.
DEFAULT VALUES	Displays the default type, label, and character string for all keys.
DISPLAY FUNCTNS	Alternately enables and disables Display Functions mode. When enabled, an asterisk is present in the label. In this mode, the action normally produced by any keyboard control or cursor control key, such as Return, Tab, or any of the display control or edit groups of keys, is not performed. Instead an ASCII character representing the function is entered in the character string definition then, when the function key is pressed in Use mode, the action is performed. The Display Functions mode used here is independent of the one enabled and disabled using either the Modes function keys, or escape sequences sent from the host computer.
define f9-f12	Displays the f9–f12 User Key menu and the associated function keys.
SAVE CONFIG	Saves the displayed parameters for keys f9–f12 in nonvolatile memory, makes the displayed parameters the active set, and returns to normal operating mode, with the System set of function keys displayed.

(continued)

Table 11-1. User Key Menu Function Keys (continued)

KEY	FUNCTION
POWER ON VALUES	Displays the parameters for keys f9–f12 stored in nonvolatile memory, which become active at power on time.
define f1-f8	Displays the f1–f8 User Key menu and the associated function keys.

The [Return] key may be used for including <CR> codes in key definitions and labels in combination with the Display Functions mode. If Auto Line Feed mode is enabled, the [Return] key will generate a <CR><LF>. If Display Functions mode is not in effect, the [Return] key will operate as a cursor movement key.

Default Definitions

Each key has a default definition. These definitions have no meaning when received by the terminal, but sent from the terminal to the computer, they can be used to trigger an action written into a program. Keys f9 through f12 have the same default definitions as keys f1 through f4.

When a hard rest is performed, or the DEFAULT VALUES function key is pressed while in Definition mode, the key types, labels (f1–f8 only), and definition strings assume their default values. At power on, the labels and definitions last saved in Definition mode, for keys f9 through f12 are displayed. Keys f1 through f8 revert to their default values, since they are not saved in nonvolatile memory.

In User Key mode, the default definition of a function key can be sent to a remote computer by pressing the [Shift] key together with the function key.

Exiting Definition Mode

To return to normal operating mode, press the [System] key; the System set of function keys will be displayed. The menu entries displayed when Definition mode is ended

automatically become the active values. For keys f9–f12, the SAVE CONFIG key can also be used to return to normal operating mode. When this key is used, the entries for keys f9–f12 are saved in nonvolatile memory. A maximum of approximately 240 characters, for all of keys f9–f12, can be saved.

The escape sequence "Ec k" may be used in place of pressing the System key. When this escape sequence is used to exit Definition mode, or when the System key is used, the System set of function key labels are displayed and their definitions are enabled. However, no entries for any of the function keys are saved in nonvolatile memory when this escape sequence is used.

Defining the User Keys Programmatically

The user keys are defined using an "`Ec &f`" sequence. The labels for keys f1–f8 can also be assigned display enhancements.

Assigning Type, Label, and Definition

From a program executing on a host computer, the user keys may be defined using the escape sequence shown below. This sequence can also be entered from the keyboard. (`<Label length>` and `<label string>` apply only to keys f1–f8.)

```
Ec &f <type> a
      <key number> k
      <label length> d
      <definition length> l
      <pair> c (color terminals only)
      <label string> <definition string>
```

where:

`<type>`	=	0:	Normal
	=	1:	Local only
	=	2:	Transmit only
`<key number>`	=	1–12:	Selects f1–f12, respectively. Default is 1.
`<label length>`	=	0 to 16:	0 is the default
`<definition length>`	=	-1 to 80:	1 is the default. 0 results in no definition string. -1 causes the field to be erased.
`<label string>`	=		The displayable character string for the label fields. The first eight characters will go into the upper half, the next eight go into the lower half.

`<definition string>` = The character string for the key definition field.

The `<type>`, `<key number>`, `<label length>`, and `<definition length>` parameters may appear in any order but must precede the label and key definition strings. Also, for color terminals, the `<pair>` parameter must follow the `<key number>` parameter. If the `<type>` parameter is not specified, it will be set to 0 (normal), although the default value is 2 (transmit only).

An uppercase identifier (A, K, D, C, or L) must be used for the final parameter and a lowercase identifier (a, k, d, c or l) for all preceeding parameters.

Following the parameters, the label string (if one is used) must precede the definition string.

Be careful to use the correct label and definition lengths. The label string is followed, in the escape sequence, by the definition string, with no boundary separator between them; the boundary is established by the label length you select.

Selection of a shorter label length than you intended will result in a truncated label and, if the definition length is greater than 0, the remaining characters of the label string will be used as the first part of the definition string. On the other hand, if the selected label length is longer than you intended, the earliest characters in the definition string will be used to finish out the label string.

Label Enhancements

The labels may have one video enhancement set selected for each label half (8 characters). If a program which creates specifications for these enhancements is to be used with 262X terminals (other than the 2625A and 2628A), these enhancements must be specified first, with a separate escape sequence. (This allows application programs to use 262X terminals which ignore unknown escape sequences.)

However, if this type terminal or a 2625A or 2628A is being used, the enhancement parameters may be included along with the other definition parameters. The label enhancement escape sequence has the following form:

```
Ec &f <key number> k
      <video enhancement> v
      <column> x
```

where the parameters mean:

`<key number>` =	1–8:	f1–f8, respectively (1 is the default)
`<video enhancement>` =	0–15:	video enhancement code, see table below for meaning. (10 is the default)
`<column>` =	1 or 9:	Column 1 selects the first half of the label to receive the enhancement; column 9 selects the second half

If the `<video enhancement>` is not specified, it defaults to inverse half-bright. Any parameter number outside of the ranges specified will cause the entire escape sequence to be ignored.

Video Enhancement Code to Enhancement Mapping

CODE	BLINKING	INVERSE VIDEO	UNDER LINE	HALF BRIGHT	NO EHNANCEMENT
0					X
1	X				
2		X			
3	X	X			
4			X		
5	X		X		
6		X	X		
7	X	X	X		
8				X	
9	X			X	
10		X		X	
11	X	X		X	
12			X	X	
13	X		X	X	
14		X	X	X	
15	X	X	X	X	

Displaying the User Key Menu Programmatically

From a program executing on a host computer or from the keyboard in Remote/Block or Local mode, the f1–f8 User Key menu may be displayed and removed using the following escape sequences:

DISPLAY f1–f8 MENU: `Ec j`
REMOVE f1–f8 MENU: `Ec k`

When removing the menu with "`Ec k`", the function key labels are displayed and the user key definitions are enabled. The "`Ec k`" enables the function keys only when the User Key menu is displayed. To programmatically enable the user keys in other circumstances, the escape sequence "`Ec &jB`" must be used.

The following escape sequences display and remove the f9–f12 menu. To display the f9–f12 menu, the f1–f8 menu must be displayed when the escape sequence is activated. When the f9–f12 menu is removed, the f1–f8 menu is displayed.

DISPLAY f9–f12 MENU: `Ec U`
REMOVE f9–f12 MENU: `Ec V`

These operations can also be performed from the keyboard. Pressing either the [Next] key or [Prev] key, while the f1–f8 menu is displayed, displays the f9–f12 menu. With the f9–f12 menu displayed, pressing either the [Prev] or [Next] key displays the f1–f8 menu.

Using the User Keys

The definition string of a user key can consist of a simple character string, an executable command (such as Display Functions, Lock Keyboard, Delete Line, etc), or both. When the definition string is executed, simple character strings are displayed on the screen and executable commands are executed. The user keys can be triggered from the keyboard or from a program.

From the Keyboard

A user key definition is executed from the keyboard by pressing the user key, but, for keys f1–f8, the terminal must be in User Key mode (keys f9–f12 are always in User Key mode). To initiate User Key mode, press the [System] and either the [Shift] or [CTRL] key, together.

From a Program

From a program executing on a host computer, the execution of any of the function keys may be triggered by using the following escape sequence:

```
Ec&f <number>E
```

where `<number>` identifies the key to be triggered. The legal values are 1 to 12, inclusive. This action is equivalent

to pressing the function key without the [Shift] key, thus triggering the user's definition; the shifted-function can not be triggered programmatically, as it serves no purpose.

When "T" type function keys are triggered (or pressed), the data string is transmitted to the computer as a data block, just as in Block mode. For information on handshaking and block-terminating characters associated with transfer of the data block, refer to Section 10.

This escape sequence may also be used within a user key definition to effectively concatenate two or more key definitions.

Controlling the User Key Labels Programmatically

From a program executing in a host computer, you can control the key labels display, in User Key mode, by using escape sequences, as follows:

`Ec &j@`	Remove all key labels from the screen. However, the user keys are still enabled.
`Ec &jA`	Display the Modes set of function key labels.
`Ec &jB`	Enable the user keys.
`Ec &jR`	Unlock screen labels.
`Ec &jS`	Lock screen labels on the screen.

Displaying a Message

You can replace the function key labels with a message you create. The escape sequence is as follows:

```
Ec&j <len>L <message>
```

where:

<len>	is the number of characters in the message. The maximum number is 160.
<message>	is the character string constituting the message.

To remove a message from the screen and display the function keys, use the following escape sequence:

```
Ec&jC
```

Programmable Return Key

The Return key is also programmable, however, this can be done only through the Terminal Configuration menu and the key definition is limited to the menu definition.

Example

The following example assigns a company name and address to key f1, to appear on the screen as follows:

```
ACME Co.
     1000 Star Rt.
     New York, NY
```

- Press the System key, then the modes key, and check whether an asterisk is present in the AUTO LF label. If so, press the associated function key to remove the asterisk.
- Press CTRL and Menu together. This initiates Definition mode and displays the User Key menu.

- Locate the cursor under the type field for [f1] and press the NEXT CHOICE key until an "L" appears in the field. This indicates the character string is for use at the terminal only.
- Move the cursor to the label line and type in your choice of label for the function key.
- Move the cursor to the left margin of the character string field.
- Press the DISPLAY FUNCTNS key to produce an asterisk in the DISPLAY FUNCTNS label.
- Type "`ACME Co.` [Return] `1000 Star Rt.` [Return] `New York, NY` [Return]".
- Press the DISPLAY FUNCTNS function key to remove the asterisk from the label. (This turns off Display Functions mode.)
- Press the [System] key, then press the modes and AUTO LF function keys to add an asterisk to the AUTO LF label. (This turns on Auto LF mode.)
- Press the [Shift] and [System] keys together to enter User Key mode. Note that the [f1] key carries your label. Press the key with your label on it. The data you typed into the character string line on the User Keys menu should appear on the screen. Note that because AUTO LF is selected, a line feed is performed following each [Return], when the function key is pressed in User Key mode.

Example

The following example uses function key f9 to automatically set the terminal into Caps Lock mode when the terminal is powered on. Similar procedures could be used to automatically perform other operations, such as logging you on to the computer system, at power on.

1. Starting with the System set of function keys displayed, in normal operating mode, press the [CTRL] and [Menu] keys together to display the f1–f8 User Key menu.
2. Press [Next] to display the f9–f12 User Key menu.
3. Press **DEFAULT VALUES**.
4. Press **NEXT CHOICE** until the Type field of f9 displays "L".
5. Press [Tab] to move the cursor to the Auto Execute field, and use the **NEXT CHOICE** key to display "Yes" in the field.
6. Press [Tab] to move the cursor to the Definition field.
7. Press the **DISPLAY FUNCTNS** key to display an asterisk in the **DISPLAY FUNCTNS** label.
8. Press the [ESC] key. Then, press the **DISPLAY FUNCTNS** key to remove the asterisk from the label.
9. Enter the string:

   ```
   &k1C
   ```
10. Press the **SAVE CONFIG** key to store the displayed user key definitions in nonvolatile memory.
11. If you are currently logged on the computer system, log off and turn off the terminal power.
12. Turn on the terminal power and enter a few alphabetic characters. They should all be displayed as upper-case characters.

Coloring the User Key Labels

For terminals with color, the user key labels can be colored using the same escape sequence used to select display enhancemnents for the labels. A color pair is selected for the background and the lettering, just as when using color in alphanumeric text. Refer to Section 4, Alphanumeric Display Control, for information on color pairs.

Note For color terminals, the default selection for the half bright enhancement is color pair 3 (yellow foreground on a black background).

Color the user key labels using the escape sequence shown below. This sequence can also be entered from the keyboard.

```
Ec &f <key>k <pair>c <column>x 0L
```

where:

`<key>`	=	1–12	Selects f1–f12, respectively. Default is 1.
`<pair>`	=	-127 to 127	Selects a color pair (0–7) for coloring the user key labels. Only the least-significant three binary bits are used. The default value is 3 if the half-bright display enhancement is selected; otherwise it is 7.
`<column>`	=	1 or 9:	Column 1 selects the first half of the label to receive the enhancement; column 9 selects the second half.

12 Designing and Using Forms

Introduction

Using the terminal's function keys, you can create a data entry form on the screen to simulate a ready-made form on the line printer, or to structure data sent to a data base. After the data entry form is created, you can read it into computer memory, using a program designed for the purpose, and incorporate it into a computer program. Then, you can use this program to print out the data entry form on the terminal screen for easy data entry.

The form for data entry is created in Local mode, using the Line Drawing character set. Titles, headings, and any other data which remains the same for all uses of the form are incorporated into the form.

A program, described and listed later in this section, can be used to copy the form and store it as a file in computer memory. You must provide a program to access or incorporate the file and display the form when you want to use it for data entry.

To expedite data entry into the form, you can initiate Format mode, in which you can tab from one field in which data is to be entered to the next. When the form contains all the desired data, you can press the [Enter] key to send the entered data to the computer. You must provide another program for receiving and processing the data.

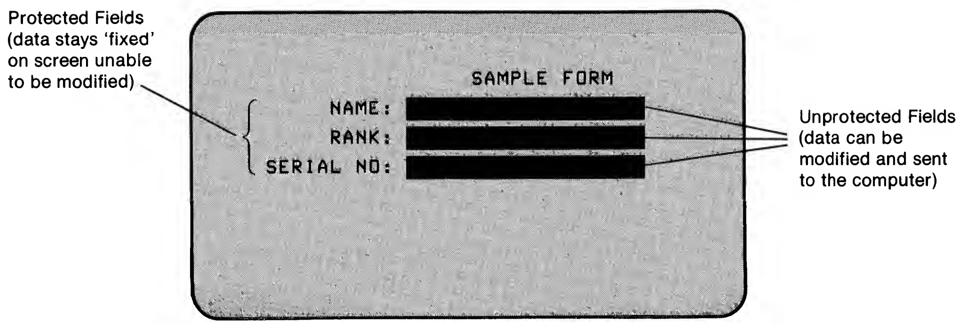
Following is a summary of the tasks involved in data entry using a form.

OPERATION	WHEN PERFORMED
Design data-entry form, on screen, in Local mode.	Once, in preparation for data entry.
Run FORMIO program to store the form in computer memory.	
Create program which will display form on screen whenever the program is run.	
Display form on screen by running the program which displays it.	Whenever data is to be entered, using the form.
Enter data into form in Format mode.	
When finished entering data, press [Enter] key, in Block, Page, and Format modes, to transmit data to computer. (A program to receive and process the data will be required.)	

Data Fields on a Data Entry Form

The data entry form can contain four types of fields: "unprotected", "protected", "transmit-only" and "security" fields. Data can be entered in unprotected and transmit-only fields and sent to the computer using the [Enter] key; data in protected fields stays fixed on the screen and cannot be modified; data entered into a security field is not displayed when it is typed in. By using the line drawing set and display enhancements to create your form, you can highlight these fields for the operator. Figure 12-1 illustrates a form with protected and unprotected fields.

Figure 12-1. Sample Form Created Using Format Mode.



Protected Data Fields

The terminal operator cannot alter or delete any characters that lie within a protected area. Protected characters are not transmitted to the host computer. The line segments and annotations that constitute the form's structure are designated as protected data.

Unprotected Data Fields

The operator enters data into unprotected fields. When the operator presses [Enter] the data in unprotected fields is transmitted to the host computer. When a character is entered into the last position of an unprotected field, the cursor automatically advances to the start of the next unprotected field. The operator may also use the Tab keys to move the cursor to the start of the next unprotected field.

Transmit-Only Fields

Transmit-only fields are similar to unprotected fields, in that they are also sent to the computer when the operator presses the [Enter] key. These fields may be modified by using the cursor control keys or commands to position the cursor in the field. (The Tab keys skip over transmit-only fields.) After reaching the end of the transmit-only field, the cursor moves to the beginning of the next unprotected field.

Transmit-only fields are desirable when you want to send fixed data such as headings or labels to the computer or when certain fields need to be modified only infrequently (e.g. dates).

Security Fields

Data entered into a security field is not displayed when it is typed in. This type field is useful for entering passwords or other security-sensitive data.

Modified Data Tags

Each field in a formatted display has a Modified Data Tag (MDT) associated with it that indicates whether or not any data has been entered into the field.

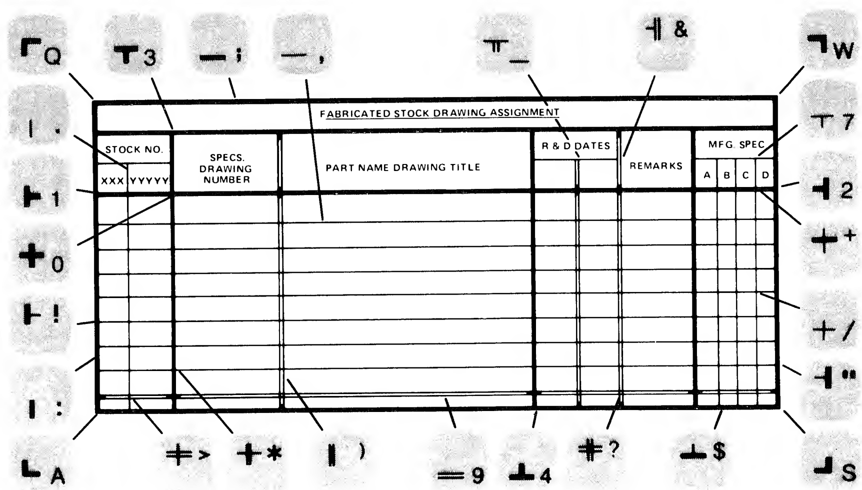
When Format mode is turned on (enabled), the MDTs for all fields in the form are automatically set "off". The entry of any valid characters into a field automatically sets the MDT for that field to "on". When one or more fields are cleared through the keyboard ([Clear display] or [Clear line]), the MDTs for the affected fields are set "on". When one or more fields are cleared programmatically (`Ec J` or `Ec K`), however, the MDTs for the affected fields are set "off".

In the Terminal Configuration menu, there is a field labeled "`Transmit Fields`" which specifies whether all fields in the form, or only those fields which have been modified, are to be transmitted to the host computer when the operator initiates a data transfer (using the [Enter] key, for example). If "`Transmit Fields=Modified`", then only those fields for which the MDT is set "on" are transmitted to the host computer. If "`Transmit Fields=All`", then all transmittable fields in the form are transmitted to the host computer regardless of how their MDTs are set.

How to Design Data Entry Forms

Data entry forms can be designed from the keyboard using the "define fields" set of function keys and the line drawing character set. Also, the "enhance video" function keys can be used to highlight the fields of the form. Figure 12-2 illustrates an example of a data entry form.

Figure 12-2. Sample Data Entry Form

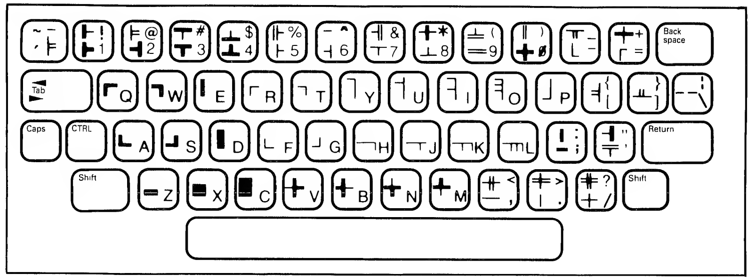


It is also possible to design forms using escape sequences, either from the keyboard or from a program.

Line Drawing Character Set

One of five character sets can be selected to be associated with the keyboard keys: the default (base) set, which consists of Roman style letters; the math set, which consists of math symbols; the line drawing set, which consists of segments of lines, useful for drawing forms; the bold set, which prints bold characters; and the italics set, which prints italics characters. All five sets are accessible through the function keys. The keyboard keys used to produce line segments are shown in figure 12-3.

Figure 12-3. Line Drawing Set Keyboard Keys



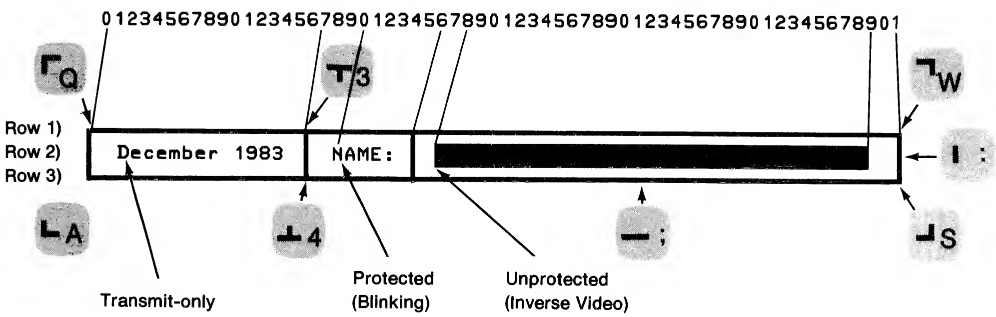
Drawing Forms Using the Function Keys

The following example illustrates the use of the line drawing character set, and the "define fields" and "enhance video" sets of function keys to create a data entry form.

Example:

Create the data entry form illustrated in figure 12-4.

Figure 12-4. Example Data Entry Form



Row 1. Press the folowing keys, in sequence:

System, enhance video, etc., CHANGE TO LINE

Draw the top line of the form, using the keys shown in figure 12-4.

Row 2. Move the cursor to the first column of row 2. Press the CHANGE TO LINE key and type ⊡.

Press CHANGE TO BASE. Move to column 3, and press the etc. and define fields keys, in sequence. Then press START XMIT FLD, type "`December 1983`", and press STOP FIELD.

Move the cursor to column 17, and display the "enhance video" function keys. Press etc., then CHANGE TO LINE, and type ⊡.

Press CHANGE TO BASE. Move the cursor to column 20, then press etc. and BLINK VIDEO. An asterisk will appear in the BLINK VIDEO key label.

Press SET ENHNCMNT. The asterisk will disappear from the BLINK VIDEO label. Then type "`NAME:`", and press SET ENHNCMNT again.

Move the cursor to column 26. Press etc., then CHANGE TO LINE, and type ⊡.

Press CHANGE TO BASE. Move the cursor to column 28, then press etc. and INVERSE VIDEO, in sequence. An asterisk will appear in the label of the INVERSE VIDEO key label.

Press SET ENHNCMNT (the asterisk should disappear from the INVERSE VIDEO label.

Press define fields, then START UNPROTCT.

Move the cursor to column 59, and press **STOP FIELD**.

Press **enhance video**, then **SET ENHNCMNT**. This ends the enhancement.

Move the cursor to column 61, press **etc.**, then **CHANGE TO LINE**, and type ⊡.

Row 3. Move the cursor to column 1 of row 3, and draw the bottom line, using the keys shown in figure 12-4.

Any change of video enhancements between the "start field" and "stop field" locations will be cleared whenever they lie within the range of a clear display or clear field operation. If you wish to define a video enhancement for an enter field which won't be erased by a "clear line" or "clear display" operation, you must do so before pressing the **START UNPROTCT** or **START XMIT FLD** function keys. Video enhancements enabled in conjunction with the start of a subfield (that is, within the overall bounds of a field) will be lost when a "clear display" or "clear field" operation is subsequently performed.

You can also define the fields and their display enhancements for the above example using the escape sequences listed in the following paragraphs in place of the function keys.

Defining Fields Programmatically

From a program executing in a host computer, you may define "unprotected" and "transmit-only" fields with the various attributes by using escape sequences. An `Ec [` specifies the start of an "unprotected" field, and `Ec {` specifies the start of a "transmit-only" field. The sequences `Ec 6`, `Ec 7`, and `Ec 8` define the various attributes of each field or subfield, and an `Ec ]` specifies the end of the field.

Refer to Section 3 for the display enhancement escape sequences.

The same sequence of operations applies when defining fields and subfields programmatically as when doing so through the keyboard. For example, if you wish the overall field to include video enhancements, you must issue the appropriate `Ec&d` sequence before issuing the `Ec [` or `Ec {` sequence. To define the start of a subfield, you issue an appropriate sequence (`Ec 6`, `Ec 7`, or `Ec 8`) at the point where the subfield is to begin.

The following escape sequences specify field types:

- `Ec 6` = begin alphabetic field (A through Z, a through z, and space only)
- `Ec 7` = begin numeric field (space, 0 through 9, minus sign, plus sign, comma, and decimal point)
- `Ec 8` = begin alphanumeric field (all keyboard characters)

How to Transfer a Form from the Screen to a Host Computer

When writing an application program that will display a form on the terminals screen, you may, of course, choose to code program statements that issue the necessary escape sequences, `SO` and `SI` codes, and data to create the form on the screen. For complex forms, however, this method can be both tedious and prone to error.

An easier method is to design the form at the terminal and then transfer it from the screen to the host computer where it can be accessed by or incorporated into a program you produce. There are two ways you might do this:

- If the terminal is connected to an HP 3000 Computer system, you may use the FORMSPEC portion of V/3000, and then include appropriate V/3000 intrinsic calls in your application program to use the form in the run-time environment.
- You can run a BASIC/3000 program called FORMIO.

The FORMIO Program

Figure 12-5 shows the source listing of FORMIO, which reads a form from the terminal screen and generates the PRINT statments necessary to recreate the form on the screen. It stores these statements as a file, which it names FDATA, and keeps as a permanent file in your account. Then it requests you to replace FORMIO with FDATA as the active program in the BASIC Interpreter (with a XEQ FDATA command), and list the statements in FDATA. Then you can modify FDATA as desired, name it, and keep it as a BASIC program. This program, when run, will reproduce the form on the terminal screen.

The file FDATA must be purged or renamed before running FORMIO again, so that FORMIO can use the file name FDATA.

FORMIO was designed primarily to assist with the programming of complex data entry forms which are much easier to create using the terminals function keys than to code directly in PRINT statements. You may, however, use it with any type of data (normal alphanumeric text, math symbols, and line drawing set elements).

Note

The following program was written to be compatible with the HP 3000 computer. If it is to be used with another type computer, modification may be necessary. In addition, the "`InhHndShk(G)`"and "`Inh DC2(H)`" fields on the Terminal Configuration menu must be set to "`NO`".

Figure 12-5. FORMIO Source Listing

```
FORMIO

  10 FILES *,*
  20 SYSTEM X1,"BUILD FDATA;rec=-160,,f,ascii"
  30 SYSTEM X1,"FILE X=$stdin;rec=-256"
  40 ASSIGN "FDATA",1,A1
  50 ASSIGN "X",2,A1,WR
  60 DIM A$[255],A1$[6],C$[3]
  70 PRINT CTL(208),'27"F"'27"a";
  80 ENTER 255,X,A$
  90 CONVERT A$[8;3] TO R
 100 PRINT "This program creates basic statements that define the"
 110 PRINT "FORM or other data in this terminals memory.";LIN(3)
 120 INPUT "Starting statement number, increment  ?",A,B
 130 PRINT CTL(208),'27"&f2a8k3L"'27";"'13'27"&f8E";
 140 LINPUT A$
 150 PRINT '27"h";
 160 PRINT #1;"scr";END
 170 FOR I=1 TO R
 180   PRINT '27"d";
 190   LINPUT #2;A$
 200   IF UPS$(A$[1,3])="RUN" THEN 500
 210   IF UPS$(A$[1,4])=">RUN" THEN 500
 220   CONVERT A TO A1$
 230   REM compensate for embedded "  marks
 240   C=-4
 250   IF C+5>LEN(A$) THEN 310
 260   C1=POS(A$[C+5],'34)
 270   IF NOT C1 THEN 310
 280   C=C1+C+4
 290   A$=A$[1,C]+"'34"+'34+A$[C+1]
 300   GOTO 250
 310   REM spaces >= 7 are converted to direct cursor addresses
 320   FOR C=1 TO LEN(A$)
 330     IF A$[C,C+6]="       " THEN DO
 340       FOR C1=C+7 TO LEN(A$)
```

Figure 12-5. FORMIO Source Listing (continued)

```
350         IF A$[C1,C1]<>" " OR LEN(A$)=C1 THEN DO
360           CONVERT C1-C TO C$
370           A$[C]='27"&a+"+DEB$(C$)+"C"+A$[C1]
380           GOTO 310
390         DOEND
400       NEXT C1
410     DOEND
420   NEXT C
430   REM output form record as a BASIC print statement
440   PRINT #1;" "+A1$+" print ctl(208),&";END
450   PRINT #1;'34+A$[1,LEN(A$) MIN 127];"&";END
460   IF LEN(A$)<128 THEN PRINT #1;'34;END
470   IF LEN(A$)>=128 THEN PRINT #1;A$[128]+'34;END
480   A=A+B
490 NEXT I
500 PRINT '27"FNow type 'XEQ FDATA' then 'LIST'.";LIN(1)
510 PRINT "These statements will reproduce your terminals memory - "
520 PRINT "modify, NAME, RENUM, and SAVE as you wish....."
530 PRINT CTL(208),'27"&f2a8k3L"'27":"'13'27"&f8E";
540 LINPUT A$
550 END
```

Using FORMIO

The sequence of events in using the FORMIO program is as follows:

- With the terminal in Local mode, clear the screen of any data you don't want to reproduce using the program to be created.
- Draw the form on the screen, using the keyboard and user keys.
- Enter Remote mode, run the BASIC Interpreter program, and call in the FORMIO program (with the GET command).
- Locate the cursor on a line below the form and above any data not to be reproduced, and enter the RUN command.

- FORMIO asks you for the starting line number and statement-numbering increment for the file of BASIC statements which it will create. For example, if you want the statements to start with number 10 and proceed in increments of 10, then enter "10,10" and press [Return].
- FORMIO reads each displayable line of display memory and creates the BASIC statement(s) necessary to reproduce each line. It keeps this list of statements as a permanent file, called FDATA, in computer memory.
- FORMIO requests you to type in "XEQ FDATA", then "LIST".
- You type in "XEQ FDATA" and press [Return].
- The BASIC Interpreter replaces FORMIO with FDATA as the program in the BASIC Interpreter workspace.
- You type in "LIST", followed by [Return].
- The list of statements comprising FDATA are listed on the screen.
- You can now modify, name, and save the FDATA statements as you would any other BASIC program you create in the BASIC Interpreter.

Entering Data into the Form

Data is entered into the form in Format mode. Forms are defined with Format mode disabled, but are not interpreted as forms by the terminal until Format mode is initiated. When Format mode is initiated, all of display memory is "protected" except for those portions which have been explicitly defined as "unprotected" and "transmit-only" fields.

The [Tab] key is used for selecting fields for data entry. In Format mode, the cursor is moved to the next unprotected field when the [Tab] key is pressed. The cursor control keys can be used to position the cursor for entering data or modifying transmit-only fields, as well as unprotected fields. Back-tabbing moves the cursor ro the previous unprotected field.

Enabling and Disabling Format Mode

The FORMAT MODE function key alternately enables and disables Format mode. When Format mode is enabled, an asterisk appears in the associated screen label. The following keystroke sequence displays that set of function keys:

System, define fields

You enable and disable Format mode programmatically by using the following escape sequences:

Enable: `Ec W`
Disable: `Ec X`

When Format mode is disabled, normal operation of the terminal is resumed.

Cursor Behavior in Format Mode

When Format mode is initiated from the keyboard, the cursor automatically moves to the start of the first unprotected field in the form (or to the "home up" position if no unprotected fields are defined). From this point on, the operator can enter data only in those portions of the display screen which lie within unprotected or transmit-only fields. When the operator enters a character into the last position of a field, the cursor advances to the start of the next unprotected field. (When the last unprotected field is filled, the cursor remains outside it and only returns to the "home up" position when the next character is entered.)

If the cursor is not within an unprotected or transmit-only field, it automatically advances to the start of the next unprotected field when the operator attempts to type a data character.

Display Control Functions in Format Mode

Tabbing. All Tab keys can be used to move the cursor forward or backward to other unprotected fields and perform any necessary scrolling. The [Tab] key wraps from the last unprotected field to the first unprotected field. The [Shift] [Tab] keys do not wrap.

Home Up/Home Down. Executing "`Ec h`" or pressing the "home up" key positions the cursor in the first unprotected field on the screen. An "`Ec H`" positions the cursor in the first unprotected or transmit-only field on the screen, whichever comes first. The Home Down function positions the cursor beneath the last line of display memory.

Cursor Control Keys, Roll Up, Roll Down. These keys may be used to position the cursor in a transmit-only field.

Clear Line/Clear Display. The Clear Line and Clear Display operations perform as follows, in Format mode:

Clear Line: If the cursor is positioned in an unprotected or transmit-only field, all displaying and non-displaying characters (except protected video enhancements) are cleared from the current cursor position through the end of the current field.

Clear Display: If the cursor is positioned within an unprotected or transmit-only field, all displaying and non-displaying characters (except protected video enhancements) are cleared from the current cursor position through the end of the current field. In addition, all subsequent unprotected fields are cleared.

Any change of enhancements between the "start field" and "stop field" locations will be cleared whenever they lie within the range of a clear display or clear line operation. If the cursor is in a protected field, these commands have no effect.

Insert Character. The insert character function affects only those characters from the cursor position through the end of the current field. If the cursor is within a protected field, it automatically moves to the first character position of the next unprotected field when the first character is inserted. The cursor moves to the next unprotected field when it reaches the last position of the current field. (When the last unprotected field is filled, the cursor remains outside it, and only returns to the "home up" position when the next character is entered.)

Delete Character. When the cursor is positioned within an unprotected or transmit-only field, the delete character function affects only those characters from the cursor position through the end of the current field. A protected video enhancement and/or an alternate character set are not altered by the delete character function. If the cursor is not within a protected or transmit-only field, the delete character function has no effect.

Insert/Delete Character with Wraparound. Insert With Wrap and Delete With Wrap will function as simple Insert and Delete functions without wrap.

Insert/Delete Line. These functions are disabled when Format mode is entered.

Margins and Tabs. All margins and tabs are cleared when the terminal enters Format mode.

Next Page/Prev Page. The cursor is moved to the first unprotected location on the new page.

How to Send Format Mode Data to a Host Computer

After all data is entered into the data entry form, you press the [Enter] key, while in Format, Block, and Page modes, to transfer the data to the host computer. Format and Block modes are initiated using the function keys; Page mode is selected on the Terminal Configuration menu.

A program must be running on the computer to receive and process the data. Only data in the transmit only and unprotected fields of the form is transmitted. For details needed to write a program to receive the data, refer to the [Enter] key discussion, as applicable to Block/Page/Format modes, in Section 10.

A Escape Codes

Introduction

Escape codes are a device which allows terminal operations to be executed from a program. When a terminal receives an escape code from an executing program, it performs the operation specified in the escape code.

Escape codes consist of most of the operations performable at the terminal using the non-alphanumeric keyboard keys. Such operations include control of the following:

- Cursor location, sensing and selection
- Display and workspaces (next page, insert line, etc)
- Margins and tabs
- Display enhancements
- User-defined keys
- Terminal modes (auto linefeed, local echo, etc)
- Alternate character set selection
- Graphics status request
- Terminal status request
- Terminal self test and reset
- Peripheral devices

All escape codes begin with the escape character "`Ec`" (produced with the [ESC] key), followed by the body of the escape code. The body of the code can consist of one or more of the keyboard letters and symbols. Most escape codes can also be performed by entering them from the keyboard. However, if the terminal is in Remote mode, the escape code will be executed only if the host computer is echoing the terminal input (the "`Local Echo`" field of the Terminal Configuration menu set to No).

Note

If the body of an escape code consists of more than one character and ends in a letter, THE TERMINATING LETTER MUST BE CAPITALIZED; otherwise, the escape code will not be recognized as such. For example, `Ec&dA` (not `Ec&da`).

To set configuration parameters using escape codes, you must use an `Ec &k`, `Ec &s`, or `Ec )` sequence, depending upon which parameters you wish to set.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
`LocalEcho`	`Ec &k`
`Caps Lock`	
`SPOW`	
`XmitFnctn(A)`	`Ec &s`
`SPOW(B)`	
`InhEolWrp(C)`	
`Line/Page(D)`	
`InhHndShk(G)`	
`Inh DC2(H)`	

The `Ec &k` and `Ec &s` sequences alter the particular parameter in the menu, but they do not alter the content of non-volatile memory.

If a configuration menu is displayed on the screen when the escape sequence is received, the sequence is stored in the terminals datacomm buffer, and is not executed until the menu is cleared from the screen.

As an example of escape code use, you can change the values of the "`Local Echo`", "`Caps Lock`", and "`SPOW`" parameters using an escape sequence of the following form:

`LocalEcho = No:`	`Ec &k 0L`
`LocalEcho = Yes:`	`Ec &k 1L`
`Caps Lock = No:`	`Ec &k 0C`
`Caps Lock = Yes:`	`Ec &k 1C`
`SPOW`	`Ec &k0N` `Ec &k1N`

You may combine these and other `Ec &k` parameters within one escape sequence. If you do, the final identifier (such as `L` or `C` or `N`) must be uppercase and all preceding identifiers must be lowercase. For example, to set `LocalEcho=Yes` and `Caps Lock=Yes`, you could use either of the following escape sequences:

```
Ec &k 1l 1C
Ec &k 1c 1L
```

TERMINAL CONTROL

KEY(S)	CODE	FUNCTION
ENTER (as used in Local mode)	`Ec 0`	Copy memory to destination(s)
System margins/ tabs/col SET TAB	`Ec 1`	Set tab
System margins/ tabs/col CLEAR TAB	`Ec 2`	Clear tab
System margins/ tabs/col CLR ALL TABS	`Ec 3`	Clear all tabs
System margins/ tabs/col LEFT MARGIN	`Ec 4`	Set left margin
System margins/ tabs/col RIGHT MARGIN	`Ec 5`	Set right margin

TERMINAL CONTROL (continued)

KEY(S)	CODE	FUNCTION
System · define fields · ONLY ALPHA	Ec 6	Define alphabetic-only field
System · define fields · ONLY NUMERIC	Ec 7	Define numeric-only field
System · define fields · ALL CHARS	Ec 8	Define unrestricted (all characters) field
System · margins/ tabs/col · CLR ALL MARGINS	Ec 9	Clear all margins
-----	Ec @	Delay one second
▲	Ec A	Cursor up
▼	Ec B	Cursor down
▶	Ec C	Cursor right
◀	Ec D	Cursor left
CTRL · Shift · Reset	Ec E	Hard reset (power on reset)
Shift · ◸	Ec F	Cursor home down
Return (with Auto LF disabled)	Ec G	Move cursor to left margin
◸	Ec H	Cursor home up
Tab	EC I	Horizontal tab
Clear display	EC J	Clear display from cursor to end of memory
Clear line	EC K	Clear display from cursor to end of line
Insert line	EC L	Insert line

TERMINAL CONTROL (continued)

KEY(S)	CODE	FUNCTION
Delete line	EC M	Delete line
Shift Insert char	Ec N	Start "Insert Character with Wraparound" mode
Shift Delete char	Ec O	Delete character with wraparound
Delete char	Ec P	Delete character (without wraparaound)
Insert char	Ec Q	Start insert character mode (insert character without wraparound)
Insert char	Ec R	End insert character
Shift ▲	Ec S	Roll up
Shift ▼	Ec T	Roll down
Next	Ec U	Next page
Prev	Ec V	Previous page
System define fields FORMAT MODE	Ec W	Format mode on
System define fields FORMAT MODE*	Ec X	Format mode off
System modes DISPLAY FUNCTNS	Ec Y	Display Functions mode on
System modes DISPLAY FUNCTNS*	Ec Z	Display Functions mode off
System define fields START UNPROTCT	Ec [	Start unprotected field

TERMINAL CONTROL (continued)

KEY(S)	CODE	FUNCTION
System define fields STOP FIELD	Ec]	End unprotected or transmit-only field
-----	Ec ^	Primary terminal status request
-----	Ec _	Write non-displaying terminator
-----	Ec `	Sense cursor position (relative)
-----	Ec a	Sense cursor position (absolute)
-----	Ec b	Unlock keyboard
-----	Ec c	Lock keyboard
-----	Ec d	Transmit a block of text to computer
-----	Ec f	Modem disconnect
Reset	Ec g	Soft reset
▷	Ec h	Cursor home up (ignoring transmit fields)
Shift Tab	Ec i	Backtab
CTRL Menu	Ec j	Display User Key Menu and begin User Key Definition mode
System or Menu	Ec k	Restore normal display and end User Key Definition mode
System modes MEMORY LOCK	Ec l	Begin Memory Lock mode
System modes MEMORY LOCK*	Ec m	End Memory Lock mode

TERMINAL CONTROL (continued)

KEY(S)	CODE	FUNCTION
f1 or f9	Ec p	Default definition for user definable function key f1 and f9
f2 or f10	Ec q	Default definition for user definable function key f2 and f10
f3 or f11	Ec r	Default definition for user definable function key f3 and f11
f4 or f12	Ec s	Default definition for user definable function key f4 and f12
f5	Ec t	Default definition for user definable function key f5
f6	Ec u	Default definition for user definable function key f6
f7	Ec v	Default definition for user definable function key f7
f8	Ec w	Default definition for user definable function key f8
System service keys DATACOMM TEST	Ec x	Initiate datacomm self-test
System service keys TERMINAL TEST	Ec z	Initiate terminal self-test
System define fields START XMIT FLD	Ec {	Start transmit-only field
-----	Ec ¦	Erase non-displaying terminator
-----	Ec ~	Secondary terminal status request

Note Columns and rows are numbered starting with 0 as the leftmost column and the top row.

CURSOR CONTROL OPERATIONS

`Ec &a <col>x <row>Y`	Moves the cursor to column "`col`" and screen row "`row`" on the screen (screen relative addressing).
`Ec &a <col>c <row>R`	Moves the cursor to column "`col`" and row "`row`" in display memory (absolute addressing).
`Ec &a +/-<col>x +/-<row>Y`	Moves the cursor to column "`col`" and row "`row`" (on the screen) relative to its present position ("`col`" and "`row`" are signed integers). A positive number indicates right or upward movement and a negative number indicates left or downward movement.
`Ec &a +/-<col>c +/-<row>R`	Moves the cursor to column "`col`" and row "`row`" relative to its present position in display memory ("col" and "row" are signed integers). A positive number indicates right or upward movement and a negative number indicates left or downward movement.

ALPHANUMERIC DISPLAY CONTROL

Ec &r <rows>D	Roll display down "rows" rows.
Ec &r <rows>U	Roll display up "rows" rows.
Ec &r <cols>L	Roll display left "cols" columns.
Ec &r <cols>R	Roll display right "cols" columns.
Ec *de	Turn on alphanumeric display.
Ec *df	Turn off alphanumeric display.
Ec *dq	Turn on alphanumeric cursor.
Ec *dr	Turn off alphanumeric cursor.
Ec &r <page>V	Display absolute vertical page "page".
Ec &r <page>H	Display absolute horizontal page "page".
Ec &r +/- <page>V	Display relative vertical page "page".
Ec &r +/- <page>H	Display relative horizontal page "page".

STATUS

Ec ^	Return terminal primary status (refer to Section 9).
Ec ~	Return terminal secondary status (refer to Section 9).
Ec *s <x>^	Returns terminal capabilities

x	CAPABILITY
-1	Alphanumeric capabilities
-2	Graphics capabilities
-3	Amount of RAM memory
-4	Interface capabilities
-5	HP-HIL Interface capabilities

Refer to Section 9 for further information.)

Ec &p <x>^	Requests the status of device "x"

x	DEVICE
4, 5, or 6	External device (printer or plotter).

MODE SELECTIONS

Ec &q 0L	Unlock configuration
Ec &q 1L	Lock configuration

These escape sequences select active values (without changing the values in non-volatile memory).

Note Only those entries in the MENU FIELD column which are marked with an asterisk are represented on a configuration menu.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &k <x>A	AUTO LF	OFF ON	x=0 x=1
Ec &k <x>B	BLOCK MODE	OFF ON	x=0 x=1
Ec &k <x>C	*Caps Lock	OFF ON	x=0 x=1
Ec &k <x>D	*Bell	OFF ON	x=0 x=1
Ec &k <x>I	*ASCII 8 Bits	NO YES	x=0 x=1
Ec &k <x>K	Auto Keyboard Lock Mode	OFF ON	x=0 x=1
Ec &k <x>L	*LocalEcho	OFF ON	x=0 x=1
Ec &k <x>M	MODIFY ALL	OFF ON	x=0 x=1
Ec &k <x>O	Numeric pad Graphics pad	----- -----	x=0 x=1
Ec &k <x>P	Caps Mode	OFF ON	x=0 x=1
Ec &k <x>Q	*Click	OFF ON	x=0 x=1
Ec &k <x>R	REMOTE MODE	OFF ON	x=0 x=1

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &k 1\	*Term Mode	ANSI	
Ec &s <x>A	*XmitFnctn(A)	NO	x=0
		YES	x=1
Ec &s <x>B	*SPOW(B)	NO	x=0
		YES	x=1
Ec &s <x>C	*InhEolWrp(C)	NO	x=0
		YES	x=1
Ec &s <x>D	*Line/Page(D)	LINE	x=0
		PAGE	x=1
Ec &s <x>G	*InhHndShk(G)	NO	x=0
		YES	x=1
Ec &s <x>H	*Inh DC2(H)	NO	x=0
		YES	x=1
Ec &s <x>J	*Auto Term(J)	NO	x=0
		YES	x=1
Ec &s <x>K	*ClearTerm(K)	NO	x=0
		YES	x=1
Ec &s <x>L	*InhSlfTst(L)	NO	x=0
		YES	x=1
Ec &s <x>N	*Esc Xfer(N)	NO	x=0
		YES	x=1
Ec &s <x>W	*InhDcTst(W)	NO	x=0
		YES	x=1
Ec &x <x>C	Send Cursor Position mode	OFF	x=0
		ON	x=1

The following table contains escape sequences which, in previous terminals, changed both the active value and the value stored in nonvolatile memory. For this terminal, only the active value is changed.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &q 8te 1{ <x>R	*RETURN=ENTER	No	x=0
		Yes	x=1
Ec &q 1te 1{ <x>G	*XmitPace	None	x=0
		XonXoff	x=1
Ec &q 1te 1{ <x>H	*RecvPace	None	x=0
		XonXoff	x=1
		TR(CD)	x=2

DATA OPERATIONS

The following escape sequences control data transfer to and from the datacomm, external device, and alphanumeric and graphics memories.

Ec &p <x>S — Selects device "x" as the source device.

x	DEVICE
3	Alphanumeric display.
5	HP-IB device.
7	Graphics display (raster dump to selected destination device).

Ec &p <x>D — Selects device "x" as the destination device, after deselecting all currently selected destination devices.

x	DESTINATION DEVICE
3	Alphanumeric display.
4, 5, or 6	External device.

Ec &p <Y> <a>d <b>d <c>D

Copies "Y" amount of data to destination devices "a", "b", and "c". As many destinations as desired can be specified.

Y	AMOUNT
b	The line in which the cursor is located.
f	From the line in which the cursor is located to the last displayed line.
m	From the line in which the cursor is located to the end of display memory.

a,b,c	DESTINATION DEVICE
3	Alphanumeric display.
4, 5, or 6	External device.

Ec &p <x>^

Requests the status of device "x".

x	DESTINATION DEVICE
4, 5, or 6	External device.

```
Ec &p    <x>p <y>C
Ec &p 4u <x>p <y>C
Ec &p 6u <x>p <y>C
```

Performs action "y" on the external device.

Note These three escape sequences are sent to the external device regardless of whether it is of the serial, Centronics or HP-IB type.

When no interface module is installed in port 2, a status of "F" will be returned.

When "4u" or "6u" is specified, the external device need not be preselected as the destination device.

If no "u" parameter is included in the escape sequence, the external device must be selected as the destination device. Otherwise a status of "F" will be returned.

y	ACTION
0	Generate a form feed.
1	Space "x" lines.
2-10	Generate a form feed.
11	Turn on Log Bottom mode.
12	Turn on Log Top mode.
13	Turn off any logging mode.
14	Print normal characters.
15	Print expanded characters.
16	Print compressed characters.
17	Turn on normal Report mode.
18	Turn on Metric Report mode.
19	Turn off any Report mode.
20	Turn on Record mode; "x" is the ASCII decimal value (1–127) used to end Record mode.

Ec &p 5u <x>p <y>C

Performs action "y" on the HP-IB device. If the HP-IB device is not installed, the command is ignored and a status of "F" will be returned. The HP-IB network need not be selected as the destination device.

y	ACTION
1	Select the device on the HP-IB bus with the HP-IB address "x" to be the object of subsequent terminal operations. (The device is normally a "talker" device.
2	Select the device on the HP-IB bus with the HP-IB address "x" to be the object of subsequent terminal operations. (The device is normally a "listener" device.)
3	Enable HP-IB device operations timeout.
4	Disable HP-IB device operations timeout.

Ec &p <x>R

Read data from the selected source device to the computer. The only valid source device for this sequence is the HP-IB device.

x	ACTION
0	Transmit next record with no byte count (default).
1	Retransmit last record only.
2	Send byte count before transmitting next record.
3	Send byte count before retransmitting last read record.
4	Transmit file.
6	Transmit file with byte count before each record.

`Ec &p <x> W <data string>`	Transfers "x" bytes of the data string from the computer to the selected destination device in binary form ("x" is a decimal value in the range 1–256).
`Ec &p W <data string>`	Transfers the data string, in ASCII form, from the computer to the printer selected as the destination device. The string is terminated either by the 256th byte or by an ASCII line feed character.
`Ec &k <x>S`	Provided the "`Protocol`" field on the External Device Configuration menu is set to "HP", sends an escape sequence to the external device to initiate Normal, Expanded, or Compressed character mode.

x	ACTION
0	Send "`Ec&k 0S`" to initiate Normal character mode.
1	Send "`Ec&k 1S`" to initiate Expanded character mode.
2	Send "`Ec&k 2S`" to initiate Compressed character mode.

FORMAT MODE

Ec [	Starts an unprotected field.
Ec {	Starts a transmit-only field.
Ec]	Ends a field.
Ec &k <x>Z	The data transmitted when the Enter key is pressed is selected by "x".

x	MEANING
0	Transmits data within the Unprotected and Transmit-Only fields (default).
1	Transmits data from only the following fields: a. Transmit-Only fields. b. Any Unprotected fields which have been modified.

FUNCTION KEY AND ERROR MESSAGE OPERATIONS

Ec &j <x>

x	MEANING
A	Display the Modes set of function key labels.
B	Enable the User function keys. (The user key labels are displayed.)
C	Disable screen messages (turn off message window and redisplay function key labels.
@	Remove the function key labels from the screen. The user keys are still enabled.
S	Disables the User System key.
R	Enables the User System key.

FUNCTION KEY AND ERROR MESSAGE OPERATIONS (continued)

```
Ec &f <attribute>a <key>k <enhancement>v <label half>x
     <label length>d <string length>l <label> <string>
```

TERM	SYMBOL	MEANING	DEFAULT
Attribute	0	Normal (N)	0
	1	Local only (L)	
	2	Transmit only (T)	
Key	1	F1 function key	1
	2	F2 function key	
	3	F3 function key	
	4	F4 function key	
	5	F5 function key	
	6	F6 function key	
	7	F7 function key	
	8	F8 function key	
	9	F9 function key	
	10	F10 function key	
	11	F11 function key	
	12	F12 function key	
Enhancement	0	None	10
	1	Blinking	
	2	Inverse video	
	3	Blinking and inverse video	
	4	Underline	
	5	Blinking and underline	
	6	Inverse video and underline	
	7	Blinking, inverse video, and underline	
	8	Half-bright	
	9	Blinking and half-bright	
	10	Inverse video and half-bright	
	11	Blinking, inverse video, and half-bright	
	12	Underline and half-bright	
	13	Blinking, underline, and half-bright	
	14	Inverse video, underline, and half-bright	
	15	Blinking, inverse video, under-line, and half-bright	

FUNCTION KEY AND ERROR MESSAGE OPERATIONS (continued)

Label half	1 9	Top half of label Bottom half of label	9
Label length	0 thru 255	Number of characters in the label. The label length plus the string length must be ≤ 255 characters. Only the first 16 characters (32 if all are muted characters) are used in the label.	0
String length	-1 thru 255	Number of characters in the string. A length of -1 clears the label. The label length plus the string length must be ≤ 255 characters. Only the first 80 characters (160 if all are muted characters) are used in the string.	1
Label	(none)	The label is entered at this point in the sequence. It may contain display enhancement and character set changes.	
String	(none)	The character string is entered at this point in the sequence. It may contain display enhancement and character set changes.	

`Ec &f <x>E` Executes the function assigned to function key "x".

x	KEY
1–12	F1–F12

`Ec &j <string length>L <message>`

"String length"—A number (up to 160) indicating the number of characters in the string.

"Message"—The content of the message.

DISPLAY ENHANCEMENTS OPERATIONS

To start and end display enhancements:

`Ec &d <char>` Selects the display enhancement indicated by "`char`" to begin at the present cursor position.

	"char"																
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	S
Half-bright									x	x	x	x	x	x	x	x	
Underline					x	x	x	x					x	x	x	x	
Inverse Video			x	x			x	x			x	x			x	x	
Blinking		x		x		x		x		x		x		x		x	
Security																	x
End Enhancement	x																

LINE ENHANCEMENTS

The following enhancements affect the entire line.

`Ec &k <num>S` — Changes the height and width of the characters in the line containing the cursor.

num	ACTION
-1	Change the characters to the top half of a double high line.
-2	Change the characters to the bottom half of a double high line.
-3	Change the characters to the top half of a double high, double wide line.
-4	Change the characters to the bottom half of a double high, double wide line.
-5	Change characters to standard size.
-6	Change characters to double width.

ALTERNATE CHARACTER SET SELECTION

`Ec )<x>` — Selects one of the secondary character sets to be the alternate set.

x	CHARACTER SET
@	Base set
A	Math set
B	Line drawing set
E	Italics
F	Bold

TEXT AND USER KEY LABEL COLOR

The following escape sequences are used with color terminals for coloring text and user key labels.

Ec &v <methd>m <fclr1>a <fclr2>b <fclr3>c <bclr1>x <bclr2>y <bclr3>z <pair>I

Assigns color values to a color pair for use in coloring text or user key labels. Any parameters missing from the sequence are set to 0. If <pair> is included in the sequence, it must follow the foreground and background values.

<methd>	0 — RGB 1 — HSL Default = 0.
<fclr1>	Red or hue value (table 4-1 or 4-2) for foreground. Default = 0.
<fclr2>	Green or saturation value (table 4-1 or 4-2) for foreground. Default = 0.
<fclr3>	Blue or luminosity value (table 4-1 or 4-2) for foreground. Default = 0.
<bclr1>	Red or hue value (table 4-1 or 4-2) for background. Default = 0.
<bclr2>	Green or saturation value (table 4-1 or 4-2) for background. Default = 0.
<bclr3>	Blue or luminosity value (table 4-1 or 4-2) for background. Default = 0.
<pair>	Identity (0–7) of the color pair for which colors are to be selected. Default = 0.

Ec &v <pair>S

Selects color pair <pair> (0–7), from the eight defined color pairs, for coloring text or user key labels.

`Ec &f <key>k <pair>c <col>k 0L`

Assigns one of eight color pairs to a user key label. The color pair is selected from the eight color pairs defined for coloring text (Section 4).

`<key>` A number (1–12) which identifies the user key.

`<pair>` A number in the range -127 to 127, which selects the color pair. Only the three least significant binary bits of the number are used. If the half bright enhancement is used, the default pair is 3; otherwise it is 7.

`<col>` Selects the label half to receive the enhancement.

1 — First half.

9 — Second half.

GRAPHICS DISPLAY CONTROL

The following escape sequence controls the graphics display.

`Ec *d <z>` Performs the indicated action "z" on the graphics display.

z	ACTION
a	Clear graphics memory
b	Set graphics memory
c	Turn on graphics display
d	Turn off graphics display
e	Turn on alphanumeric display
f	Turn off alphanumeric display
k	Turn on graphics cursor
l	Turn off graphics cursor
m	Turn on rubber band line
n	Turn off rubber band line
<x,y> o	Move graphics cursor to horizontal position "x" and vertical position "y" (relative to the origin)
<x,y> p	Move graphics cursor to horizontal position "x" and vertical position "y" (relative to its present location)
q	Turn on alphanumeric cursor
r	Turn off alphanumeric cursor
s	Turn on graphic text mode
t	Turn off graphics text mode
z	No operation

GRAPHICS DISPLAY CONTROL (continued)

Ec *d <Lx><Ly> <Hx><Hy>y	Select display resolution.

Lx	Ly	Hx	Hy	
0	0	511	389	Low resolution
0	0	639	399	High resolution

GRAPHICS TEXT

Ec *m <x>m — Sets the graphics text size to "x", where "x" is a number from 1 to 8.

Ec *m <x>n — Sets the graphics text orientation to "x".

x	ROTATION (degrees)
1	0
2	90
3	180
4	270

Ec *m o — Turns on text slant.

Ec *m p — Turns off text slant.

Ec *m <x>q — Sets the origin of graphics text at location "x" on the display.

GRAPHICS TEXT (continued)

x	LOCATION
0	left/baseline
1	left/bottom
2	left/middle
3	left/top
4	center/bottom
5	center/middle
6	center/top
7	right/bottom
8	right/middle
9	right/top
10	center/baseline
11	right/baseline

This escape sequence is used for transmission of a graphics text label from a program to the terminal.

`Ec *l <text>` `CR, CR LF,` `LF CR, or LF`	The characters contained in "text" are printed on the display starting at the pen position

RELOCATABLE ORIGIN

`Ec *m k`	Locates the relocatable origin at the current pen position.
`Ec *m l`	Locates the relocatable origin at the graphics cursor position.

GRAPHICS STATUS

This escape sequence reads the graphics status.

`Ec *s <x>^`	Reads status type "x".

x	STATUS
1	Terminal I.D.
2	Pen position
3	Graphics cursor position
4	Read cursor position and wait for key
5	Display size
6	Graphics capabilities
7	Graphics text status
8	Read zoom status
9	Relocatable origin
10	Reset status
11	Area shading
12	Dynamics
32	Graphics positioning device connected
33	Requests current cursor position (provided the tablet or mouse originates the transfer).

GRAPHICS DEFAULTS

`Ec *m r`	Set graphics defaults:

PARAMETER	DEFAULT
* Pen Condition	Down
* Line Type	1 (solid)
* Drawing Mode	2 (JAM—monochrome) 2 (JAM 1—color)
* User Defined Line Pattern	255, 1
* Area Fill Type	2 (User Defined Pattern)
* User Defined Area Fill Pattern	255, 255,..., (Solid)

GRAPHICS DEFAULTS (continued)

* Background Pen	0 (Black)
* Primary Pen	7 (White)
* Secondary Pen	0 (Black)
* Boundary Pen	Off
* Graphics Text	Off
* Text Size	1
* Text Direction	1
* Text Origin	1 (left, bottom)
* Text Slant	Off
* Text Color	Primary Pen
Relocatable Origin	0,0
Alpha Video	On
Graphics Video	On
Alpha Cursor	On
Graphics Cursor	Off
Graphics Cursor Address	0,0
Rubberband Line	Off
Compatibility Mode:	
Page Full Straps	0 (Out)
GIN Strap	0 (CR Only)

Note Parameters marked with an asterisk are those affected by the sequence "`Ec *m <1>r`".

`Ec *m <1>r` Sets the graphics defaults which are marked with an asterisk in the list above.

`Ec *m z` No operation.

DRAWING LINES

The following escape sequences are used to draw lines.

Ec *m <x>a	Selects drawing mode "x" for monochrome terminals.

x	MODE
0	No effect
1	Clear
2	Set
3	Complement
4	Jam

Ec *m <x>b	Selects line type "x".

x	LINE TYPE
1	Solid line
2	User line pattern
3	Current area pattern
4	Line #1
5	Line #2
6	Line #3
7	Line #4
8	Line #5
9	Line #6
10	Line #7
11	Point plot

Ec *m <x><y>c

Defines an eight-bit segment of line pattern and a scale; where:

"x" is a number from 0 to 255 which, when converted to its binary form, illustrates the segment of line pattern.

"y" is a number from 0 to 255 which indicates the number of times the line pattern should be repeated.

Ec *p <x>	Performs action "x".

x	ACTION
a	Lift the pen
b	Lower the pen
c	Use graphics cursor position as new point
d	Draw a point at the current pen position and lift the pen
e	Set relocatable origin at the current pen position
f	Data is ASCII absolute
g	Data is ASCII incremental
h	Data is ASCII relocatable
i	Data is binary absolute
j	Data is binary short incremental
k	Data is binary incremental
l	Data is binary relocatable
s	Start area fill
t	End area fill
u	Lift area boundary pen
v	Lower area boundary pen
z	No operation

`Ec *m <a b c d e f g h>d`

Defines an 8 x 8 pattern where "a" through "h" are numbers from 0 through 255 which, when converted to their binary values and stacked, illustrate the pattern.

`Ec *m <x1,y1,x2,y2,>e`

Defines a rectangular area to be filled, where "x1,y1" and "x2,y2" define the rectangle located with respect to the absolute origin.

`Ec *m <x1,y1,x2,y2>f`

Defines a rectangular area to be filled, where "x1,y1" and "x2,y2" define the rectangle with respect to the relocatable origin.

`Ec *m <x,y>j`

Locates the relocatable origin at coordinates "x,y" with respect to the absolute origin.

`Ec *m <x>g`

Selects area pattern "x":

x	AREA PATTERN
1	Solid area fill.
2	User-defined area fill (default).
3	Predefined pattern 0 (short dashed hatching).
4	Predefined pattern 1 (long dashed hatching).
5	Predefined pattern 2 (hatching).
6	Predefined pattern 3 (cross hatching).
7	Predefined pattern 4 (fine cross hatching).
8	Predefined pattern 5 (medium checkerboard).
9	Predefined pattern 6 (fine checkerboard, 1:1 blend).
10	Predefined pattern 7 (3:1 blend).

`Ec *m <x>h`

Select area boundary pen "x"; where "x" is an integer in the range -32767 through 32767. The three low bits of the binary form of the integer is used to select the pen. Any selection other than 0, including no parameter specified, enables the area boundary pen.

COMPATIBILITY MODE

These escape sequences are used in Compatibility mode.

Ec *t <x>a — Selects graphics terminator.

x	TERMINATOR
0	CR
1	CR EOT
2	None

Ec *t <x>b — Sets or clears Page Full Break strap.

x	ACTION
0	Clear
1	Set

Ec *t <x>c — Sets or clears Page Full Busy strap.

x	ACTION
0	Clear
1	Set

Ec *t <x>d — Sets or clears 4014 mode:

x	ACTION
0	4010 mode
1	4014 mode

Ec *t z — No operation.

Ec *w r — Graphics hard reset.

The color graphics escape sequences are listed below.

`Ec *v <x>P` — Selects palette <x>.

`Ec *v <method>m <int>a <int>b <int>c <mix>I` — Defines the color for one mix.

- <method> 0 = RGB (default)
  1 = HSL
- <int> Intensity for each of the three color components. Values can range from 0.00 to 1.00, but are reduced to one of the following RGB values: 0.00, 0.33, 0.66, or 1.00. (HSL values are converted to RGB values.)

  a—Red or hue.

  b—Green or saturation.

  c—Blue or luminosity.
- <mix> The mix to which the color is to be assigned.

`Ec *v L` — Loads the selected palette. (The palette becomes palette 0, the system palette.

`Ec *e <x>B` — Assigns mix <x>, on palette 0, to the background pen.

`Ec *m <x>X` — Assigns mix <x>, on palette 0, to the primary pen.

`Ec *m <x>Y` — Assigns mix <x>, on palette 0, to the secondary pen.

`Ec *n <x>X` — Assigns mix <x>, on palette 0, to the graphics text pen.

`Ec *m <x>H` — Activates the area boundary pen and assigns mix <x>, on palette 0, to it.

`Ec *m H` — Disables the area boundary pen.

`Ec *m <x>A` — Selects the drawing mode.

<x>	MODE
0	Picture Protect
1	Clear1
2	Jam1
3	Complement1
4	Jam2

5	OR
6	Complement2
7	Clear2

`Ec *m <x>W`

Selects predefined dithered color <x> as the current dithered color.

x	COLOR
2	Violet
3	Brown
4	Burnt orange
5	Gold
6	Lime
7	Turquoise
8	Red
9	Green
10	Blue
11	White
12	Black
13	Lime green
14	Aqua
15	Royal blue
16	Fuchsia
17	Burnt sienna
18	Pumpkin
19	Gray Brown

`Ec *m <x>,<y>,<z>V`

Assigns densities <x>, <y>, and <z> to the red, green, and blue color planes to select a user-defined dithered color. <x>, <y>, and <z> are values in the range 0.00–1.00, which are converted by the terminal to the nearest 1/16th (0.06, 0.13, 0.19, 0.25, etc.).

`Ec *v <mask> <mix> <pal>^`

Returns palette/mix status information.

<mask> A decimal number in the range -32768 to 32767 which is converted to a binary number which selects status requests from the list which follows. Only the six least significant binary bits are used. If the number is 0, no

information is returned.

A "1" selects and a "0" deselects a request.

If a palette is specified, a mix must also be specified, or the palette number will be interpreted as the mix number, resulting in erroneous information.

MIX BIT	REQUEST	RESPONSE
0	Palette exist?	1 = yes 0 = no
1	Is palette the currently loaded palette?	1 = yes 0 = no
2	Color selection method?	0 = RGB 1 = HSL
3	No. of currently loaded palette?	+nnnnn (0–15)
4	Mix RGB values?	R = n.nnnn, G = n.nnnn, B = n.nnnn
5	Mix HSL values?	H = n.nnnn, S = n.nnnn, L = n.nnnn

`<mix>` The number of the mix for which information is requested when bit 4 or 5 is set. If none is specified, information for mix 7 is returned.

`<pal>` The number of the palette for which information is requested when one or more of bits 0, 1, 4, or 5 is set. If any of these bits is set with no palette specified, information for the system palette (palette 0) is returned.

`Ec*j 0A`	Sets input device off line.
`Ec*j 1A` (default)	Sets input device on line; input device modes (as listed below) set to default values.
`Ec*j 0B` (default)	Selects Low Resolution mode (data scaled to screen resolution).
`Ec*j 1B`	Selects High Resolution mode (data scaled to four times screen resolution).
`Ec*j 2B`	Selects Raw Data mode (unscaled data from tablet).
`Ec*j 2B`	Selects Raw Data mode (actual data from input device).
`Ec*j 0C` (default)	Selects Synchronous Data Transfer mode.
`Ec*j 1C`	Selects Asynchronous Data Transfer mode.
`Ec*j 0D` (default)	Selects Point Pen mode.
`Ec*j 1D`	Selects Pen Down mode.
`Ec*j 2D`	Selects Pen Toggle mode.
`Ec*j 0E` (default)	Selects ASCII data format.
`Ec*j 1E`	Selects binary data format.
`Ec*j 0F` (default)	Releases the "pick" operation of the input device from assignment to any function key.
`Ec*j <key>F`	Assigns the "pick" operation of the input device to the function key `<key>` (9–12).

TABLET/MOUSE STATUS

Ec*s 3^	Read cursor position immediately.
Ec*s 4^	Read cursor position if the input device initiates a data transfer or a keyboard key is pressed.
Ec*s 32^	Requests input device identification.
Ec*s 33^	Requests cursor position only if input device originates a data transfer.

TOUCHSCREEN CONTROL

Ec-z 4n	Toggles touchscreen on and off.
Ec-z 5n	Select unprotected fields to be touch sensitve for cursor positioning by touch. (Default)
Ec-z 7n	Switch from Alphanumeric to Touch Mouse mode.
Ec-z 0n	Disable all reporting except any function keys which have been defined as touch keys.
Ec-z 1n	Row/column reporting only. Touch field reporting disabled.
Ec-z 2n	Touch field reporting only. Row/column reporting disabled.
Ec-z 3n	Both row/column and touch field reporting.
Ec-z 6n	Normalized reporting only.
Ec-z 1m	Selects the touch condition to trigger the report for row/column reporting.
Ec-z 2m	Selects the release condition to trigger the report for row/column reporting.

(continued)

TOUCHSCREEN CONTROL (continued)

`Ec-z 3m`	Selects both the touch and the release conditions to trigger a report for row/column reporting.
`Ec-z <row>x <col>y 3Q`	Report escape sequence sent to host computer program by a touch when row/column reporting is selected. Report triggered on touch.
`Ec-z <row>x <col>y 4Q`	Report escape sequence sent to host computer program by a touch when row/column reporting is selected. Report triggered on release.
`Ec-z <xcoord>x <ycoord>y 3Q`	Report escape sequence sent to host computer program by a touch when normalized reporting is selected. Report triggered on touch. `<xcoord>` and `<ycoord>` are the x and y coordinates (0.000 to 1.000).
`Ec-z <xcoord>x <ycoord>y 4Q`	Report escape sequence sent to host computer program by a touch when normalized reporting is selected. Report triggered on release. `<xcoord>` and `<ycoord>` are the x and y coordinates (0.000 to 1.000).

DEFINE A TOUCH FIELD

```
Ec-zg <rows>r <cols>c <beep>b <enh1>e <enh2>f <enhr>h  <enhc>i <box>t
      <type>a <tcond>m <cpos>p <slen>L <string>
```

where:

`<rows>`	Two row numbers, separated by a comma, which define the upper and lower bounds of the touch field. The default is 0,47.
`<cols>`	Two column numbers, separated by a comma, which define the left and right bounds of the touch field. The default is the configured workspace width -1.

<beep>

0 – Disable audible beep on activation of touch field (default selection).

1 – Enable beep on activation of touch field.

<enh1>

A number, 0–31, which selects the enhancement for the off state of the touch field. The default is no enhancement. The enhancement need not be confined to the touch field. (In the table below, "sec" is an abbreviation for "security".)

sec off	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
sec on	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
half-brt									x	x	x	x	x	x	x	x
undrline					x	x	x	x					x	x	x	x
invs vid			x	x			x	x			x	x			x	x
blinking		x		x		x		x		x		x		x		x

<enh2>

Same as <enh1>, except it selects the enhancement for the on state of the touch field. The default is no enhancement.

<enhr>

Two row numbers, separated by a comma, which define the upper and lower bounds of an enhancement field. This field would normally (but not necessarily) be within the touch field defined by "<rows>r" and "<cols>c". The enhancement is of the type selected by "<enh1>e" and "<enh2>f". These row numbers default to the rows selected by "<rows>r."

<enhc>

Two column numbers, separated by a comma, which define the left and right bounds of the enhancement field described under "<enhr>". These column numbers default to column numbers selected by "<cols>c".

<box>

0 – No box drawn around the touch field.

1 – A box is drawn around the touch field.

(continued)

<type>

A number, 1, 3, or 4, which selects the type of the field:

1 – ASCII field. (Default selection.) When the touch field is activated, the report string of the field is transmitted to the host computer.

3 – Toggle field. When the touch field is activated, the report string of the field and information about the state of the field is transmitted to the host computer.

For each activation of the touch field, the enhancement toggles (from on to off, and vice-versa). This allows a toggle-type field to be used like a switch, with one enhancement indicating the on condition and the other the off condition.

4 – Normal field. When the touch field is activated, the report string of the field and information about the state of the field is transmitted to the host computer.

<tcond>

A number, 1–3, which selects the touch condition which triggers the report for the touch field:

1 – Report on touch only (default selection).

2 – Report on release only.

3 – Report on both touch and release.

<cpos>

A number, 0–1, which selects whether or not to move the cursor to the upper left corner of the touch field when the field is touched:

0 – Do not position cursor.

1 – Position cursor.

<slen>

A number, 0–80 for ASCII fields and exactly 2 for Toggle and Normal fields, which indicates the number of characters in the report string. The default string length is 0.

`<string>`	The report string, which is sent to the host computer when the touch field is activated. For ASCII fields, it can be up to 80 characters of any type. For Toggle and Normal fields, it must be two lower-case ASCII characters. These characters identify the touch field to the host computer program.
`Ec-z <key>s <res>k`	Select/de-select a function key to be activated by a touch on the label.
`<key>`	Identity of the function key, 1–8 (default=1).
`<res>`	`0` – Disable touch response. `1` – Enable touch response.
`Ec-z <row>r <col>cD`	Delete touch field whose upper left corner is identified by `<row>` and `<col>`.
`Ec-zD`	Delete all touch fields.
`Ec-zJ`	Reset all touch fields to the off state.
`Ec-z <string> 1Q`	The report escape sequence sent to the host computer program when a Toggle-type touch field is switched on. `<string>` identifies the touch field.
`Ec-z <string> 2Q`	Same as above, except the report is sent when the touch field is switched off.
`Ec-z <string> 5Q`	The report escape sequence sent to the host computer program when a Normal-type touch field is touched.
`Ec-z <string> 6Q`	Same as above, except the report is sent when the touch field is released.

ANSI ESCAPE SEQUENCES

CONTROL CHARACTERS

CONTROL CHARACTER	HEX CODE	TERMINAL ACTION
NUL	00	Ignored on input.
ENQ	05	Transmits an answerback message (or ACK if the "EnqAck" field in the Datacomm Configuration menu is set to "Yes").
BEL	07	Sounds the bell.
BS	08	Moves the cursor one position to the left, stopping at the left margin of current line.
HT	09	Moves the cursor to the next tab stop, stopping at right margin if no further tab stops are in the line.
LF	0A	Executes a linefeed or a new line operation (see new line mode, LNM).
VT	0B	Interpreted as a LF.
FF	0C	Interpreted as a LF.
CR	0D	Moves cursor to left margin of current line.
SO	0E	Invokes alternate character set, as defined by select character set sequence.
SI	0F	Invokes the base character set, as define by the select character set sequence.
XON	11	Causes terminal to resume transmission.
XOFF	13	Causes terminal to stop transmitting all characters except XON and XOFF.

CONTROL CHARACTERS (continued)

CAN	18	If sent during a control sequence, the sequence is immediately terminated and not executed.
SUB	1A	Interpreted as CAN.
ESC	1B	Introduces a control sequence.
DEL	7F	Ignored as input.

CURSOR MOVEMENT COMMANDS

Ec [Pn A	Cursor Up
Ec [Pn B	Cursor Down
Ec [Pn C	Cursor Forward
Ec [Pn D	Cursor Backward
Ec [Pn; Pn H	Cursor Position
Ec [Pn; Pn f	Horizontal and Vertical Position
Ec [Pn G	Cursor Horizontal Absolute
Ec [Pn E	Cursor Next Line
Ec [Pn F	Cursor Preceding Line
Ec [Pn Z	Cursor Back Tab
Ec [Pn `	Horizontal Position Absolute
Ec [Pn a	Horizontal Position Relative

CURSOR MOVEMENT COMMANDS (continued)

`Ec [ Pn d`	Vertical Position Absolute
`Ec [ Pn e`	Vertical Position Relative
`Ec D`	Index (cursor down)
`Ec E`	Next Line
`Ec M`	Reverse Index
`Ec 7`	Save cursor attributes
`Ec 8`	Restore cursor and attributes
`Ec [ > 1 s`	Home down
`Ec [ > 0 s`	Home up

DISPLAY CONTROL SEQUENCES

`Ec [ Pn U`	Next Page
`Ec [ Pn V`	Preceding Page
`Ec [ Pn S`	Scroll Up
`Ec [ Pn T`	Scroll Down
`Ec [ Pn <space>@`	Scroll left
`Ec [ Pn <space>A`	Scroll right

LINE SIZE

Ec # 3	Make top half of this line double high, double wide.
Ec # 4	Make bottom half of this line double high, double wide.
Ec # 5	Make this line standard size.
Ec # 6	Make this line double wide.

CHARACTER ATTRIBUTES

Ec [Ps;...;Ps m	Defines character ehancements.

Ps	MEANING
0 or none	All attributes off
1	Half-bright on
4	Underline on
5	Blink on
7	Inverse video on

EDITING COMMANDS

Ec [Pn P	Delete Character
Ec [Pn M	Delete Line
Ec [Pn L	Insert Line

ERASING

Ec [Ps K	Erase in line.

Ps	DEFINITION
0	Erase from cursor to end of line
1	Erase from start of line to cursor, inclusive
2	Erase entire line

Ec [Ps J	Erase display.

Ps	DEFINITION
0	Erase from cursor to end of display
1	Erase from start of screen to the cursor position, inclusive
2	Erase all of the display

CHARACTER SETS

Ec (Ps	Character set defined to be base set:

Ps	CHARACTER SET
B	Configured language
0	ANSI Line Drawing
3	HP Line Drawing

Ec) Ps	Character set defined to be alternate set:

Ps	CHARACTER SET
B	Configured language
0	ANSI Line Drawing
3	HP Line Drawing

TABS

Ec [Pn; Pn r	Sets top and bottom of scrolling region
Ec H	Horizontal Tab Set
Ec [Ps g	Clears tab stops

Ps	DEFINITION
0 or none	Clear tab at current column
3	Clear all tabs

MODES

Ec [Pc;...;Pc l	Resets the terminal modes defined by the parameters Pc. The available parameters are those Pc numbers listed in the table below.
Ec [Pc;...;Pc h	Sets the terminal modes defined by the parameters Pc. Again, the available parameters are those Pc numbers listed in the table below.

ESCAPE SEQUENCE	MODE	ENTRY VALUE	Pc
Ec [2 Pc	Keyboard Action	Locked	Pc=h
		Unlocked	Pc=l
Ec [3 Pc	Control Representation	Display	Pc=h
		Exec fnc	Pc=l
Ec [4 Pc	Insert/Replace	Insert	Pc=h
		Replace	Pc=l
Ec [12 Pc	Send receive mode	Echo Off	Pc=h
		Echo On	Pc=l

ESCAPE SEQUENCE	MODE	ENTRY VALUE	Pc
Ec [20 Pc	Line feed/new line	Newline Linefeed	Pc=h Pc=l
Ec [38 Pc	TEK mode		Pc=h
Ec & k Pc \	HP mode	HP	Pc=0
Ec <	From EM52 to ANSI	ANSI	-----
Ec [?1 Pc	Cursor keys mode	Applic Cursor	Pc=h Pc=l
Ec [?2 l	ANSI to EM52 mode	EM52	-----
Ec [?3 Pc	Column Mode	132 80	Pc=h Pc=l
Ec [?4 Pc	Scrolling Mode	Smooth Jump	Pc=h Pc=l
Ec [?6 Pc	Origin Mode	Relative Absolute	Pc=h Pc=l
Ec [?7 Pc	Wraparound	On Off	Pc=h Pc=l
Ec = Ec >	Keypad Mode	Applic Numeric	----- -----
Ec [> 0 Pc	User key menu	On Off	Pc=h Pc=l
Ec [> 1 Pc	Multipage Mode	Set Reset	Pc=h Pc=l
Ec [> 2 Pc	Memory Lock Mode	Set Reset	Pc=h Pc=l

REPORT REQUESTS

`Ec [ 5 n`	Device Status Report
`Ec [ 6 n`	Cursor Position Report
`Ec [ 0 c` or `Ec [ c` or `Ec Z`	Device Attributes
`Ec [Ps x`	Request terminal parameters. `Ps` can have the value `0` or a `1`. Parameters are reported using the report terminal parameters sequence.

REPORT REPONSES

`Ec [ 0 n`	Terminal Ready, no errors
`Ec [ 3 n`	Malfunction, retry
`Ec [ Pn; Pn R`	Cursor Position Report
`Ec [ ?1;2c`	Device Attributes (VT100)

`Ec [ 3;<par>;`
`<nbits>;`
`<xspeed>;`
`<rspeed>;`
`1;0x`
	Report terminal parameters. See the following table for values. Note that, on this terminal, terminal transmit speed (`<xspeed>`) and terminal receive speed (`<rspeed>`) are always the same.

PARAMETER	VALUE	MEANING
`<par>`	1	No parity set
	2	Space parity
	3	Mark parity
	4	Parity set to Odd
	5	Parity set to Even
`<nbits>`	1	8 bits per character
	2	7 bits per character

REPORT REPONSES (continued)

<xspeed> and <rspeed>	16	110 Baud
	32	150 Baud
	48	300 Baud
	56	600 Baud
	64	1200 Baud
	72	2400 Baud
	88	4800 Baud
	112	9600 Baud
	120	19200 Baud

MEDIA COPY

`Ec [ Ps i`

Controls the transfers between the terminal and the external device provided the terminal is fitted with a second port) according to the parameter.

Ps	DEFINITION
0	Copy All
5	Turn on Passthrough mode
4	Turn off Passthrough mode

`Ec [ ? Ps i`

Turns on/off Log Bottom mode.

Ps	DEFINITION
5	Turn on Log Bottom mode
4	Turn off Log Bottom mode

RESET

`Ec c`

Hard Reset

LOAD FLAGS

`Ec [ Ps;...q`	Controls the four programmable flags in the display status line.

Ps	EFFECT
0	Clear all flags
1	Set flag "L1"
2	Set flag "L2"
3	Set flag "L3"
4	Set flag "L4"

VT52 ESCAPE SEQUENCES

CURSOR MOVEMENT

`Ec A`	Cursor up
`Ec B`	Cursor down
`Ec C`	Cursor right
`Ec D`	Cursor left
`Ec H`	Home cursor
`Ec I`	Reverse line feed
`Ec Y<row><col>`	Position cursor

CHARACTER SET

`Ec F`	Select and enable alternate character set
`Ec G`	Select and enable base character set

ERASING

Ec J	Erase to end of screen
Ec K	Erase to end of line

MODES

Ec <	Enter ANSI mode.
Ec =	Enter Alternate Keypad mode
Ec >	Exit Alternate Keypad mode
Ec F	Enter Graphics (line drawing) mode.
Ec G	Exit Graphics mode.

IDENTIFICATION

Ec Z	Identify — request transmitted by host program
Ec/Z	Identify — response transmitted by terminal

B ANSI/EM52 Operation

Introduction

The 2393A terminal can operate on computer systems using ANSI (American National Standards Institute) X3.64 protocol. ANSI Operation allows the terminal to respond to ANSI escape sequences, changes functions of certain keys, and provides the terminal with answerback capability.

The following is a summary of the terminal's ANSI Operation features:

1. Hewlett-Packard (HP) mode of operation: the terminal operates as a standard HP terminal. It uses the HP capabilities, communications and escape sequences described in the other sections of this manual.
2. ANSI mode of operation: the terminal recognizes and executes a sub-set of the terminal escape sequences specified in the American National Standards Institute documents X3.41-1974 and X3.64-1979. It also recognizes and executes some of the standard HP parameterized escape sequences. In ANSI mode:
   a. An ANSI Configuration menu allows the user to define:
      - The terminal "answer back" message (response to a computer enquiry, "ENQ").
      - Tab settings, which are saved in the terminal's non-volatile memory.
      - The [Back space] key operation (the key can be used as a backspace or a delete key).
      - The size of display memory.

b. Keys ◄, ►, ▲, ▼ can be programmed to operate as cursor control keys or "Application mode" keys.

c. The function of some of the keys in the numeric pad is changed. A "numeric pad overlay", supplied with the terminal, indicates the function of these keys. Also, the keys in the numeric pad can be programmed to operate as numeric keys or "applications mode" keys.

d. A terminal mode indicator ("ANSI") and four programmable application flags are available in the Status Line.

e. Line drawing characters are available.

3. EM52 mode: the terminal emulates a VT52® compatible terminal by recognizing and executing escape sequences defined for a DEC VT52® terminal.

In EM52 mode, the ANSI Configuration menu is available to:

- Select an "answer back" message.
- Define tab settings.
- Define the [Back space] key operation.

The function of some of the keys in the numeric pad is changed (the same as for ANSI mode). Also, the keys in the numeric pad can be programmed to operate as numeric keys or "applications mode" keys. A terminal mode indicator ("EM52") is displayed in the Status Line. Note that Status Line flags are not available, and keys ◄, ►, ▲ and ▼ can only operate as cursor control keys.

® VT52 and DEC are registered trademarks of Digital Equipment Corporation.

ANSI/EM52/HP Mode Configuration

ANSI mode adds an additional field to the Global Configuration menu (figure 6-1). This field, labeled "Term Mode", can be set to "HP" (default), "EM52" or "ANSI". The setting of this field determines the terminal's mode of operation. The default setting is "HP".

An ANSI Configuration menu is available for selection of the following items in ANSI or EM52 mode:

a. Answerback message (up to 40 characters).

b. Size of display memory: one page or multiple pages.

c. [Back space] key operation (backspace/delete character).

d. Tab stops (which are saved in non-volatile memory).

Refer to Section 6 for detailed configuration information.

Modes of Operation

The terminal has three operating modes: HP, EM52, and ANSI. At power-on, the mode is determined by the setting in the Global Configuration menu. After power-on, the mode can be changed programmatically using escape sequences, or by reconfiguring the terminal. Note that programmatically changing the mode of operation does not alter the contents of the non-volatile memory (which sets the operating mode at power-on or after a hard reset).

HP Mode

In HP mode, the terminal performs all the functions of a standard HP terminal. It responds to the escape sequences and performs all the terminal functions described in all other sections of this manual. The terminal does not respond to either VT52® or ANSI escape sequences. The center of the status line contains an "HP" when the terminal is in HP mode (figure B-1).

Figure B-1. HP Mode Screen Status Line

device control	margins/ tabs/col	service keys	modes	21 44 112 234	enhance video	define fields		config keys
KB Lockd	Ext Char	Tab=Spac	Num Pad or Grph Pad	* HP *	CAPS	Ins Char or Ins Wrap	TouchOff	STOP

EM52 Mode

In EM52 mode, the terminal emulates a VT52® compatible terminal. The terminal responds to the escape sequences written for a DEC VT52® terminal (listed at the end of this section) and to selected HP sequences (listed later).

In EM52 mode, a terminal mode indicator "EM52" is displayed in the Status Line, and the terminal provides some of the features available in ANSI mode (figure B-2).

Figure B-2. EM52 Mode Screen Status Line

device control		service keys	modes	21 44 112 234	enhance video	define fields		config keys
KB Lockd	Ext Char	L1L2L3L4	Num Pad or Grph Pad	* EM52 *	CAPS	Ins Char	TouchOff	STOP

In ANSI/EM52 mode, the terminal is automatically set to Character mode and the associated function key is disabled. However, Line Modify and Modify All modes can be used in both EM52 and ANSI modes.

Note

Line Modify and Modify All modes, transmission of "transmit type" user key strings, and graphics status requests may require DC1/DC2 handshaking, depending on the "InhHndShk(G)" and "Inh DC2(H)" field selections on the Terminal Configuration menu.

The functions of some of the keys in the numeric pad are changed:

a. Keys [*], [/], [+] and [-] act as program function keys [PF1] to [PF4].

b. [Enter] acts as [Return] in Application mode.

c. [Tab] acts as [Enter].

The "numeric pad overlay", supplied with the terminal, indicates the function of these keys. The keys in the numeric pad (except [PF1] to [PF4]) can be programmed to operate as numeric pad keys or "applications mode" keys.

The terminal may use any of the parameters configurable in the ANSI Configuration menu. These are:

a. Answerback message (up to 40 characters).

b. Size of display memory: one page or multiple pages.

c. [Back space] key operation (backspace/delete character).

d. Tab stops (which are saved in non-volatile memory).

These features are described later in this section.

Note that, in EM52 mode, the four application flags in the Status Line are not available, keys [◄], [►], [▲] and [▼] can only be used as cursor control keys, and the [Insert char], [Delete char], [Insert line], and [Delete line] keys are disabled.

ANSI Mode

ANSI is the mode of operation that gives the terminal the ability to operate with a computer system using ANSI control sequences. In ANSI mode, the terminal responds to the ANSI escape sequences described at the end of this section and the same HP escape sequences as in EM52 mode. It does not respond to VT52® escape sequences.

In ANSI/EM52 mode, the terminal is automatically set to Character mode (asterisk removed from BLOCK MODE function key label), and the associated function key is disabled. However, Line Modify and Modify All modes can be used.

Note

Line Modify and Modify All modes, transmission of "transmit type" user key strings, and graphics status requests may require `DC1/DC2/DC1` handshaking, depending on the "`InhHndShk(G)`" and "`Inh DC2(H)`" field selections on the Terminal Configuration menu.

In ANSI mode, a terminal mode indicator "`ANSI`" is displayed in the status line (figure B-2). Four application flags (L1, L2, L3 and L4) in the status line can be turned on and off programmatically. The function of some of the keys in the numeric pad is changed:

- Keys [*], [/], [+] and [-] act as program function keys [PF1] to [PF4].
- [Enter] acts as [-], and [Tab] as [Enter].

Figure B-3. ANSI Mode Screen Status Line

device control		service keys	modes	21 44 112 234	enhance video	define fields		config keys
KB Lockd	Ext Char	L1L2L3L4	Num Pad	* ANSI *	CAPS	Ins Char	TouchOff	STOP

The "numeric pad overlay", supplied with the terminal, indicates the function of these keys. The same overlay is used for both ANSI and EM52 mode.

The keys in the numeric pad (except [PF1] to [PF4]) can be programmed to operate as numeric pad keys or "applications mode" keys. Keys ◄, ►, ▲, ▼ can be programmed to operate as cursor control keys or "Application mode" keys.

The [Insert char], [Delete char], [Insert line], and [Delete line] keys are disabled in ANSI mode.

A line drawing character set is available.

In ANSI mode, the terminal can transmit a user-definable answerback message by pressing the [CTRL] and [Break] keys, together. Also, a long break (3.5 seconds) can be transmitted to the host computer by pressing the [CTRL] and [STOP] keys together.

In addition, the terminal may use any of the parameters configurable in the ANSI Configuration menu. These are:

- Answerback message (up to 40 characters).
- Size of display memory: one page or multiple pages.
- [Back space] key operation (backspace/delete character).
- Tab stops (which are saved in non-volatile memory).

These features are detailed later in this section.

Escape Sequences

The terminal recognizes different escape sequences according to the operating mode. If the mode is changed, any previously received escape sequences that are not recognized in the new mode are lost. For instance, if the terminal was in ANSI mode with a scrolling region defined, and the mode is changed to HP, the scrolling is ignored (the scrolling region is NOT re-instated if the terminal is subsequently re-set to ANSI mode).

HP Mode

When the terminal is configured for HP mode, it responds to the escape sequences listed in Appendices A, and does not respond to either VT52® or ANSI escape sequences.

EM52 Mode

When the terminal is configured for EM52 mode, it responds to the escape sequences written for a DEC VT52® terminal listed at the end of this section and to the following HP mode escape sequences. Refer to the HP portion of Appendix A for details on these escape sequences.

SEQUENCE	OPERATION
Ec&f	Define and execute function keys.
Ec&j	Enable, disable, and display function keys and error messages.
Ec&k	Select terminal modes (exception: Ec&k<x>B - set and clear Block mode).
Ec&q	Lock/unlock configuration, equate [Return] key to [Enter] key, and select transmit/receive pacing.
Ec&s	Select terminal modes.
Ec-z	Touchscreen control and touchfield definition.
Ec*	All graphics operations.

Note

When the `Ec-zg` escape sequence is used to define a touchfield, the `<enh1>`, `<enh2>`, `<enhr>`, `<enhc>`, and `<box>` parameters must not be included in the sequence; if they are, the sequence will be treated as invalid.

ANSI Mode

When the terminal is configured for ANSI mode, it responds to the ANSI compatible escape sequences found at the end of this section. It also responds to the same HP sequences responded to in EM52 mode.

The terminal does not respond to VT52 escape sequences.

The ANSI sequences are a subset of those specified in Standards ANSI X3.41-1974 and ANSI X3.64-1979. Those sequences designated as private by ANSI and ISO code extension standards (ANSI X3.41-1974 and ISO 2022 1973, respectively) are not yet standardized. Therefore, Hewlett-Packard has assigned functions to some of these ANSI compatible sequences, and prefixed their mnemonic with "HP". All the other mnemonics used are the same as those specified in the ANSI standards.

Escape Sequences Generated by the Keyboard

Escape sequences are also generated at the keyboard by active edit keys (such as [Clear line] and [Clear display]), system-level function keys (such as [MEMORY LOCK]), and cursor control keys. The transmission of these escape sequences from the terminal to the host computer depends on the terminal mode:

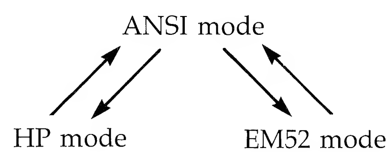
- In HP mode, the escape sequences are only transmitted to the host computer when the "`XmitFnctn(A)`" field in the Terminal Configuration menu (figure 6-2, Section 6) is set to "Yes".

- In ANSI mode, escape sequences generated by both the numeric keypad keys (when set to Application mode) and the cursor control keys are always transmitted, and the `XmitFnct(A)` field is ignored. For the remaining keys, the escape sequences are only transmitted when the "`XmitFnctn(A)`" field is set to "`YES`".
- In EM52 mode, only escape sequences generated by the numeric keypad keys (when set to Application mode) and the cursor control keys are transmitted. The "`XmitFnctn(A)`" field is ignored.

Non-transmitted escape sequences are executed locally (at the terminal). Transmitted escape sequences are only executed locally if the computer performs a remote echo or the terminal's local echo is ON.

Selecting the Terminal Mode Programatically

The terminal mode can be configured programmatically from the host computer, but only in the following directions:



The escape sequences used to configure the mode are as follows:

1. HP mode. HP mode can only be selected when currently operating in ANSI mode, using ANSI escape sequence:

   `Ec&k0\` (from ANSI mode to HP mode)
2. EM52 mode. E52 mode can only be selected when currently operating in ANSI mode, using ANSI private escape sequence:

   `Ec[?2l` (from ANSI mode to EM52 mode)

3. ANSI mode. ANSI mode can be selected when operating in either HP or EM52 mode using:

a. HP escape sequence:

`Ec&k1\` (from HP mode to ANSI mode)

b. EM52 escape sequence:

`Ec<` (from EM52 mode to ANSI mode)

Cursor Control Keys

In HP and EM52 mode, the four keys ◄, ►, ▲ and ▼ operate as cursor control keys, and generate the characters listed in Table B-1. In ANSI mode (only), the four keys ◄, ►, ▲ and ▼ can be operated as:

1. Cursor control keys. These keys generate control sequences that control the movement of the cursor (see Table B-2).

2. "Application mode" keys. These keys generate different control sequences (see Table B-2); the functions performed depend on the application program.

The cursor key Application mode is set or reset programmatically using the following ANSI escape sequences:

SET (=Application mode):	`Ec[?1h`
RESET (=cursor control mode):	`Ec[?1l`

Tables B-1 and B-2 list the control sequences generated by each key.

The keys revert to cursor control keys at power-on or after a hard reset.

Table B-1. HP And EM52 Control Sequences Generated by Cursor Keys

Key	HP Mode	EM52 Mode
▲ Cursor up	EcA	EcA
▼ Cursor down	EcB	EcB
▶ Cursor right	EcC	EcC
◀ Cursor left	EcD	EcD

Table B-2. ANSI Control Sequences Generated by Cursor Keys

Key	Cursor Key Application mode RESET (cursor movement keys)		Cursor Key Application mode SET (application mode keys)	
▲	Ec[A	Cursor up	EcOA	Application function
▼	Ec[B	Cursor down	EcOB	
▶	Ec[C	Cursor right	EcOC	
◀	Ec[D	Cursor left	EcOD	

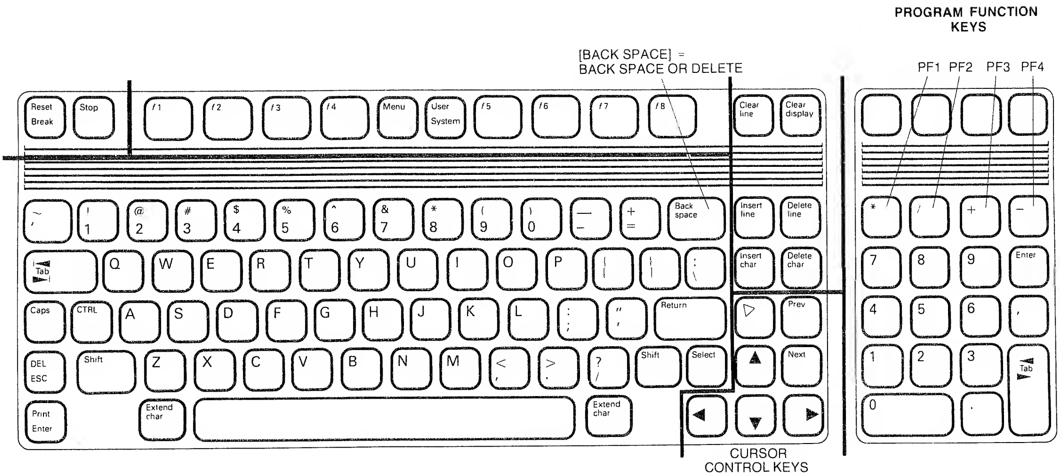
Numeric Keypad

When the terminal is set for ANSI or EM52 operations, the numeric pad is "re-mapped" with:

- Keys [*], [/], [+], and [-] used as program function keys [PF1] through [PF4].
- [Enter] as [Return].
- [Tab] as [Enter].
- Keys [0] to [9] are unchanged.

The numeric pad overlay, supplied with the terminal (figure B-4) indicates the function of these keys. The same overlay is used for both ANSI and EM52 modes.

Figure B-4. Numeric Pad Overlay



The numeric keypad has two modes of operation: Numeric mode and Application mode.

In Numeric mode, the keys operate as re-mapped above. In other words, keys in the numeric keypad (except [PF1] to [PF4]) transmit the same codes as the corresponding keys on the main keyboard (see Table B-3). Therefore, the host computer cannot tell if these keys were pressed on the numeric keypad or on the main keyboard.

Table B-3. ANSI and EM52 Control Sequences Generated By Numeric Keypad

Numeric Keypad Overlay Legend	CONTROL SEQUENCE SENT TO HOST COMPUTER			
	ANSI		EM52	
	Numeric Mode	Application Mode	Numeric Mode	Application Mode
0	0	ESC O p	0	ESC ? p
1	1	ESC O q	1	ESC ? q
2	2	ESC O r	2	ESC ? r
3	3	ESC O s	3	ESC ? s
4	4	ESC O t	4	ESC ? t
5	5	ESC O u	5	ESC ? u
6	6	ESC O v	6	ESC ? v
7	7	ESC O w	7	ESC ? w
8	8	ESC O x	8	ESC ? x
9	9	ESC O y	9	ESC ? y
-	-	ESC O m	-	ESC ? m
.	.	ESC O n	.	ESC ? n
,	,	ESC O l	,	ESC ? l
[Enter]	Same as [Return]	ESC O M	Same as [Return]	ESC ? M
PF1	ESC O P	ESC O P	ESC P	ESC P
PF2	ESC O Q	ESC O Q	ESC Q	ESC Q
PF3	ESC O R	ESC O R	ESC R	ESC R
PF4	ESC O S	ESC O S	ESC S	ESC S

In Application mode, the keys in the numeric pad (except [PF1] to [PF4]) generate different control sequences (see Table B-3), the interpretation of which depends on the application program.

Function keys [PF1] to [PF4] generate the same control sequences in both numeric and Application mode (see Table B-3). The use of these keys depends on the application program.

The numeric keypad Application mode is set or reset programmatically using the following escape sequences:

	ANSI Mode	EM52 Mode
SET (=Application mode):	Ec =	Ec =
RESET (=Numeric mode):	Ec >	Ec >

Table B-3 lists the control sequences generated by each key.

The keypad reverts to Numeric mode at power-on or after a hard reset.

Alternate Character Sets

EM52 Mode

In EM52 mode, the terminal has two character sets: base and alternate. These are selectable either programmatically or from the keyboard, as follows:

Ec F — select alternate character set
Ec G — select base set

ANSI Mode

In ANSI mode, the terminal has two character sets: base and alternate. These sets are selectable from the host or keyboard, using the following ANSI escape sequence:

Ec (Ps — select base character set (G0)
Ec) Ps — select alternate character set (G1)

Where Ps can have the values:

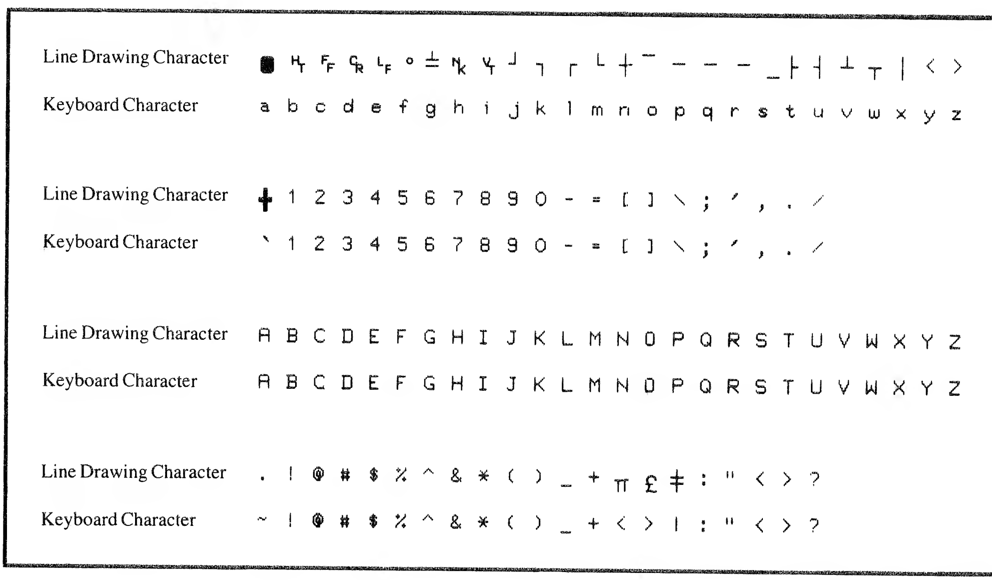
B = configured language.
0 = ANSI line drawing set.

At power-on or after receipt, the default base character set (G0) is always the configured language set. The default alternate character set (G1) is always the ANSI line drawing set.

To invoke the base set (previously selected by the appropriate ANSI escape sequence) while operating in the alternate set, issue a Shift-in (SI) control character.

The alternate set (G1) is enabled by the Shift-out (SO) control character. The alternate set stays enabled until you perform a hard reset or send a Shift-in (SI) control character.

Figure B-5. ANSI Line Drawing Set Elements



ANSI Configuration Menu

When the terminal is set for ANSI mode or EM52 mode, the ANSI Configuration menu (figure 6-8) is available to define:

1. The size of the display memory.
2. The terminal "answer back" message (response to a computer enquiry, ENQ).
3. Tab settings, which are saved in the terminal's non-volatile memory.
4. The [Back space] key operation, which can be used as both a backspace and a delete key.

To display the ANSI Configuration menu, first ensure that the terminal is in ANSI or EM52 mode, then press the keys in the following order:

[System], **config keys**, **ansi config**

Multipage Field

This is a multipage terminal which dynamically stores data. Applications programs designed for ANSI and EM52 terminals often require the terminal to have only a single page of memory (24 lines). Consequently, this terminal can be configured to use a single page or multiple pages. When a single page is used, keys [Next], [Prev], roll up ([Shift] [▲]) and roll down ([Shift] [▼]) are disabled.

To change the "MultiPage" parameter in the ANSI Configuration menu:

1. Place the cursor in the field with either the [Tab] key or the cursor control keys. Use the **NEXT CHOICE** or **PREVIOUS CHOICE** function keys to change the field to the desired choice.
2. Save the new configuration by pressing the **SAVE CONFIG** function key. This causes the System labels to be displayed. The saved configuration becomes the active configuration, and is saved in non-volatile memory.

In ANSI mode, the multipage parameter can be set and reset programmatically using the following ANSI escape sequences:

SET (=multipage):	`Ec[>1h`
RESET (=single page):	`Ec[>1l`

Changing the value of this parameter will clear display memory and home up the cursor. The multipage parameter value set programmatically is not saved in the nonvolatile memory. If the power is switched OFF, the terminal reverts to the value in the "`Multipage`" field.

Backspace Definition Field

In ANSI/EM52 mode, the [Back space] key has two functions:

- Backspace—generates a `BS` control code (hexadecimal 08) which moves the cursor one character backwards (leftwards) along the line, stopping at column 1. This is a non-destructive backspace function.
- Delete—generates a `DEL` control code (hexadecimal 7F) which is usually interpreted by the host computer to mean "delete the preceding character and move the cursor one space left".

The "`BackspaceDef`" field in the ANSI Configuration menu specifies the operation of the [Back space] key when pressed by itself (unshifted) and when pressed with [Shift] (shifted). To change the "`BackspaceDef`" parameter:

1. Place the cursor in the field with either the [Tab] key or the cursor control keys. Use the **NEXT CHOICE** or **PREVIOUS CHOICE** function keys to change the field to the desired choice (backspace/delete or delete/backspace).
2. Save the new configuration by pressing the **SAVE CONFIG** function key. This causes the System labels to be displayed.The saved configuration becomes the active configuration, and is saved in non-volatile memory.

The operation of the [Back space] key is not programmable from the host computer.

Setting and Saving Tabs

Tab stops defined in the ANSI Configuration menu are saved in non-volatile memory, and are not lost if the terminal is powered off. Note that these tab stops are only available in ANSI/EM52 mode; they are ignored in HP mode. To set ANSI tab stops:

1. Place the cursor in the desired position in the tab stop indicator line (at the bottom of the menu). Then use the NEXT CHOICE or PREVIOUS CHOICE key to toggle on and off the tab stop (a stop is indicated by a "T"). The label CLR ALL TABS clears all the tab stops in all columns in the menu (except for the implicit tab at column 1).
2. Press key [f1] (SAVE CONFIG) to store the tab stops, and return the display to the System labels. The saved tab stops become the active tab stops and are saved in non-volatile memory.

In HP and ANSI modes, tab stops can be set and cleared programmatically using the following escape sequences (in EM52 mode, tab stops cannot be set/cleared programmatically):

	HP Mode	ANSI Mode
SET tab stop:	`Ec1`	`EcH`
CLEAR tab stop:	`Ec2`	`Ec[0g` or `Ec[g`
CLEAR ALL tab stops:	`Ec3`	`Ec[3g`

The tab stops are not saved in the non-volatile memory. If the power is switched OFF (or a hard reset is performed), the terminal reverts to the values in non-volatile memory.

Answerback Message Field

In ANSI/EM52 mode, the terminal has the ability to identify itself to the host computer with a user-defined character string called an answerback message. The answerback message is transmitted when the terminal receives an "ENQ" character (hexadecimal 05) from the computer.

When the ANSI Configuration menu is first displayed, the "Anwerback Message" field does not appear.
This field is only displayed when the ANSWER BACK key is pressed. In addition (for security), the configured answerback message cannot be recalled once it has been defined.

By default no message exists, and it is up to the user to define the message. This can be done as follows:

1. Press key [f5] (ANSWER BACK) to display the "Answerback Message" parameter. The message field (displayed in half-bright inverse video) is blank, with the cursor at the start of the field. You can then enter a message of up to 40 characters in the field.
2. Ensure the answerback message field is displayed, then press SAVE CONFIG (key [f1]). This causes the System labels to be redisplayed and all the displayed menu parameters to be saved. The saved answerback message becomes the active message, and is saved in non-volatile memory (replacing the previously stored message). If the answerback message is not displayed when SAVE CONFIG is pressed, the previously stored message is retained.

Note that the answerback message is not programmable from the host computer. The answerback message can be triggered from the keyboard by pressing the [CTRL] and [Break] keys, together.

ANSI Control Sequence Summary

The control sequences recognized and processed by the terminal when in ANSI mode are detailed below.

Terms. The following terms apply in ANSI mode:

Active position:	The position of the cursor.
Control sequence:	A string of characters that is used to perform a special function, that contains the Control Sequence Introducer (`ESC [`), some parameters which may be optional, and a final character. If a sequence contains several selective or numeric parameters, successive parameters must be separated by semicolons ";" (hexadecimal 3B).

Numeric parameter (`Pn`):	A string of numbers which represents a numeric value between two range limits.
	
	For example, when moving the cursor with a CUF (Cursor Forward) sequence:
	`ESC [ Pn C`
	
	Pn is a numeric parameter that can have any value between 1and 80. If the cursor is to be moved 7 places to the right, the sequence is:
	`ESC [ 7 C.`

Scrolling region:	The part of the display which rolls up when new data is received by the terminal. If no top or bottom margins are in effect, the scrolling region is the whole display.
Selective parameter (Ps):	A string of characters which selects one function among a list of several functions. The selective parameter, designated as Ps, may ONLY have a value which is chosen from the list of functions. Any other value will be interpreted as an error.

Control Characters

Control characters (hex 00 to 1F inclusive) are excluded from the control sequence syntax, but may be embedded within a control sequence. Embedded control characters are executed as soon as they are received by the terminal, provided display functions mode is not set. The processing of the control sequence then continues with the next character encountered.

Table B-4 lists the control characters processed by the terminal in ANSI or EM52 mode.

Table B-4. ANSI/EM52 Mode Control Character Recognition

CONTROL CHARACTER	HEX CODE	TERMINAL ACTION
NUL	00	Ignored on input
ENQ	05	Transmits an answerback message, if any exists.
BEL	07	Sounds the bell.
BS	08	Moves the cursor one position to the left, stoipping at column 1.
HT	09	Moves the cursor to the next tab stop, stoppping the last column if no further tab stops are in the line.
LF	0A	Executes a linefeed or a new line operation (see new line mode, LNM).
VT	0B	Interpreted as LF.
FF	0C	Interpreted as LF.
CR	0A	Moves cursor to column 1 of current line.
SO	0E	Invokes G1 character set, as defined by SCS sequence (see select character set sequence).
SI	0F	Invokes G0 character set, as defined by SCS sequence.
X-ON	11	Causes terminal to resume transmission.
X-OFF	13	Causes terminal to stop transmitting all characters except X-ON and X-OFF.
CAN	18	If sent during a control sequence, the sequence is immediately terminated and not executed. It also causes the error character ([]) to be displayed.
SUB	1A	Interpreted as CAN.
ESC	1B	Introduces a control sequence.
DEL	7F	Ignored on input.

ANSI Cursor Control Sequences

The cursor control escape sequences responded to or transmitted by the terminal in ANSI mode are listed in tables B-5 and B-6.

Table B-5. ANSI Specified Sequences

MNEMONIC	DESCRIPTION
CBT	Cursor Back Tab
CHA	Cursor Horizontal Absolute
CNL	Cursor Next Line
CPL	Cursor Preceding Line
CPR	Cursor Position Report
CUB	Cursor Backward
CUD	Cursor Down
CUF	Cursor Forward
CUP	Cursor Position
CUU	Cursor Up
HTS	Horizontal Tab Set (opposite to TBC)
HVP	Horizontal and Vertical Position (same as CUP)
IND	Index (similar to CUD)
NEL	Next Line (similar to CNL)
RI	Reverse Index (similar to CUU)
TBC	Tab Clear
VPA	Vertical Position Absolute
VPR	Vertical Position Relative
HPA	Horizontal Position Absolute (same as CHA)
HPR	Horizontal Position Relative (same as CUF)

Table B-6. HP Private Escape Sequences

MNEMONIC	DESCRIPTION
HPHD	Home Down
HPHU	Home Up
HPSC	Save Cursor and attributes
HPRC	Restore Cursor and attributes

These sequences are described below.

CBT — Cursor Back Tabulation

`Ec [ Pn Z`

Moves the cursor horizontally backward along the active line to the Pnth preceding tab stop. `Pn = 0` or `1` moves the cursor to the first preceding tab position.

CHA — Cursor Horizontal Absolute

`Ec [ Pn G`

Moves the active cursor position forward or backward along the active line to the specified column position. `Pn = 0` or `1` moves the cursor to the first position in the active line. A parameter value of `Pn` moves the cursor to the Pnth column of the active line. A value greater than the display capacity moves the cursor to the right edge of the display.

The CHA sequence has the same effect as the HPA (Horizontal Position Absloute) sequence.

CNL — Cursor Next Line

`Ec [ Pn E`

Moves the active cursor position to the first column of the Pnth subsequent line. `Pn = 0` or `1` indicates the next line. A parameter value of Pn moves the cursor down by `Pn` lines. If line `Pn` is below the last displayed line, a roll up is performed (if permitted).

CPL — Cursor Preceding Line

`Ec [ Pn F`

Moves the active cursor position to the first position of the Pnth previous line. `Pn = 0` or `1` indicates the previous line. A parameter value of Pn moves the cursor up by `Pn` lines. If line `Pn` is above the first line, a roll down is performed (if required).

CPR — Cursor Position Report (terminal to host)

`Ec [ Pn; Pn R`

The CPR sequence is generated in response to a DSR (Device Status Report) sequence requesting a cursor position report. The CPR sequence reports the active cursor position by means of the two parameters:

- The first specifies the line.
- The second specifies the column.

CUB — Cursor Backward

`Ec [ Pn D`

Moves the cursor to the left by the specified number of columns. `Pn` = `0` or `1` or no parameter, moves the cursor by one position. The cursor stops when column 1 is reached.

CUD — Cursor Down

`Ec [ Pn B`

Moves the cursor down by the specified number of screen lines. `Pn` = `0` or `1` or no parameter, moves the cursor down by one line. If an attempt is made to move the cursor past the bottom margin or line 24, it will stop there (no rolling is performed).

CUF — Cursor Forward

`Ec [ Pn C`

Moves the cursor to the right by the specified number of columns. `Pn` = `0` or `1` or no parameter, moves the cursor right by one column. The cursor stops at the right edge of the screen.

CUP — Cursor Position

`Ec [ Pn; Pn H`

Moves the cursor to the specified position. This sequence has two parameters:

- The first specifies the line number.
- The second specifies the column number.

Line numbering depends on the set/reset state of the Origin mode (HPOM):

If the HPOM (Origin mode) is set, the cursor can only be positioned within the margins of the scrolling region defined using HPSTBM, and the lines are numbered with respect to the first line of the scrolling region.

If both parameters are 0 or no parameters are specified, the cursor moves to the home up position.

The CUP sequence has the same effect as the HVP (Horizontal and Vertical Position) sequence.

CUU — Cursor Up

`Ec [ Pn A`

Moves the cursor up by the specified number of screen lines (without changing the column number). `Pn` = `0` or `1` or no parameter, moves the cursor up by one line. If an attempt is made to move the cursor past the top margin, it will stop there (no rolling is performed).

HTS — Horizontal Tab Set

`Ec H`

Sets one tab stop at the active cursor position.

If the user subsequently displays the ANSI Configuration menu and presses SAVE CONFIG, this tab stop plus any others set in the menu will be stored in nonvolatile memory.

These tab stops are only effective in ANSI/EM52 mode. They are ignored if the terminal is set to HP mode, but are re-instated if the terminal is subsequently re-set to ANSI or EM52 mode.

HVP — Horizontal and Vertical Position

`Ec [ Pn; Pn f`

Moves the cursor to the specified position.

The HVP operates the same way as the CUP (Cursor Position) sequence.

IND — Index

`Ec D`

Moves the active cursor position down by one line (without changing the column number). If an attempt is made to move the cursor past the bottom margin or line 24, a roll up is performed (if required). If "Auto LF" is ON, then the active cursor position moves to column 1.

NEL — Next Line

`Ec E`

Moves the active cursor position to the first position on the next line (downward). If an attempt is made to move the cursor below the bottom margin or line 24, a roll up is performed (if required).

RI — Reverse Index

`Ec M`

Moves the active cursor position up by one line (without changing the column number). If an attempt is made to move the cursor above the top margin, a roll down is performed (if required).

TBC — Tab Clear

`Ec [ Ps g`

Clears the tab position according to the parameter:

Ps=0 or none	Clear tab at active position.
Ps=3	Clear all horizontal tabs

VPA — Vertical Position Absolute

`Ec [ Pn d`

Moves the cursor to the specified line (without changing the horizontal position). A parameter of `Pn` moves the cursor vertically to line `Pn`. If there is no Pnth line, the cursor moves to the last available line.

VPR — Vertical Position Relative

`Ec [ Pn e`

Moves the cursor downward by the specified number of lines (without changing the horizontal position). `Pn` = `0`, or `1` or no parameter, moves the cursor one line down. If line `Pn` is below the last displayed line, a roll up is performed (if required).

HPA — Horizontal Position Absolute

``Ec [ Pn ` ``

Moves the active cursor position forward or backward along the active line to the specified column position.

HPR — Horizontal Position Relative

`Ec [ Pn a`

Moves the cursor to the right by the specified number of columns.

HPHD — Home Down

`Ec [ > 1 s`

Performs a Home Down, text is rolled up as required. This is equivalent to pressing keys [Shift] [⇱].

HPHU — Home Up

`Ec [ > 0 s`

Performs a cursor Home Up, text is rolled down as required. This is equivalent to pressing the [⇱] key.

HPSC — Save Cursor And Attributes

`Ec 7`

Causes the active cursor position, base/alternate character sets and display enhancement to be stored in temporary memory, where they are available for subsequent retrieval using the HPRC sequence. Note that the stored values are lost when the terminal is powered off or after a hard reset.

HPRC — Restore Cursor And Attributes

`Ec 8`

Recalls the values stored by the last HPSC (Save Cursor) sequence. Note that no action is taken if an HPSC sequence has not been received since the terminal was last powered-up or reset.

ANSI Display Control Sequences

The following display control sequences are responded to by the terminal in ANSI mode:

	Mnemonic	Description
ANSI specified sequences	NP	Next Page
	PP	Previous Page
	SD	Scroll Down
	SU	Scroll Up
HP private sequence	HPSTBM	Set top and bottom margin

NP — Next Page

`Ec [ Pn U`

Causes the Pnth subsequent page of data in the display memory to be displayed. `Pn` = `0` or `1` or no parameter, displays the next page. A parameter value of `Pn` displays the Pnth subsequent page. The new page is displayed with the cursor in the home up position.

Note that the terminal ignores the NP sequence:

1. If the `Multipage` field in the ANSI Configuration menu is `NO` (one page).
2. If a scrolling region of less than 24 lines is defined using HPSTBM.

PP — Previous Page

`Ec [ Pn U`

Causes the Pnth preceding page of data in display memory to be displayed. `Pn` = `0` or `1` or no parameter, displays the previous page. A parameter value of `Pn` displays the Pnth previous page The new page is displayed with the cursor in the home up position.

The terminal ignores the PP sequence if the `Multipage` field in the ANSI Configuration menu is NO (one page), or if a scrolling region of less than 24 lines is defined using HPSTBM.

SD — Scroll Down

`Ec [ Pn T`

Causes all the data displayed on the screen to be moved down by the specified number of lines. `Pn` = `0` or `1` or no parameter, scrolls the data down one line. A parameter value of `Pn` scrolls down `Pn` lines.

The cursor remains in the same position on the screen. As the bottom line of data is removed from the screen, another new line appears at the top. Scrolling stops when the first line on the screen is the first line in the display memory.

The HPSCLM (HP Scrolling mode) sequence selects jump or smooth scrolling.

SU — Scroll Up

`Ec [ Pn S`

Causes all the data displayed on the screen to be moved up by the specified number of lines. `Pn` = `0` or `1` or no parameter, scrolls the data up one line. A parameter value of `Pn` scrolls up `Pn` lines.

The cursor remains in the same position on the screen. As the top line of data is removed from the screen, another new line appears at the bottom. If scrolling continued until the bottom of display memory is reached, the last line in the memory is displayed at the top of the screen.

The HPSCLM (HP Scrolling mode) sequence selects jump or smooth scrolling.

Scroll Left

```
Ec [ Pn <space> @
```

Scroll Right

```
Ec [ Pn <space> A
```

HPSTBM — Set Top and Bottom Margins

```
Ec [ Pn; Pn r
```

At power-on, by default, the scrolling region is the entire screen. The HPSTBM sequence allows the scrolling region to be set to between 2 and 24 lines by means of two parameters:

1. The first specifies the line number of the first line of the scrolling region, and can have a value from 1 to 23.
2. The second specifies the line number of the bottom line of the scrolling region, and can have a value from 2 to 24.

The parameter values are screen lines, and are included in the scrolling region. The top/bottom margins are not saved in non-volatile memory. They are preserved between ANSI and EM52 modes.

The minimum size of the scrolling region is two lines (and the top margin must have a line number less than than the bottom margin). The cursor is placed in the home position. (Refer to Origin mode, HPOM.)

A parameter value of 0,0 sets the scrolling region equal to the entire screen.

ANSI Editing Control Sequences

The terminal responds to the following editing control sequences in ANSI mode:

	Mnemonic	Description
ANSI specified sequences	DCH	Delete Character
	DL	Delete Line
	ED	Erase in Display
	EL	Erase in Line
	IL	Insert Line

DCH — Delete Character

```
Ec [ Pn P
```

Deletes the specified number of characters at the active cursor position. `Pn` = `0` or `1` or no parameter deletes one character. Whenever a character is deleted, all characters to the right of the active cursor position (in the current line) are moved one character position left.

DL — Delete Line

```
Ec [ Pn M
```

Deletes the specified number of lines at the active cursor position. `Pn` = `0` or `1` or if no parameter, delete one line. Whenever a line is deleted, all the lines below the deleted line are moved one line up. The sequence is ignored when the cursor is outside the scrolling region.

ED — Erase in Display

```
Ec [ Ps J
```

Erases some or all of the displayed characters according to the parameter:

- `Ps=0` Erase from cursor to the end of the display (default).
- `Ps=1` Erase from start of screen to cursor, inclusive.
- `Ps=2` Erase all of the display.

The ED sequence does not move the cursor.

EL — Erase in Line

```
Ec [ Ps K
```

Erases some or all the characters in the active line according to the parameter:

- `Ps=0` Erase from cursor to the end of line, inclusive (default).
- `Ps=1` Erase from start of line to cursor, inclusive.
- `Ps=2` Erase entire line.

The EL sequence does not move the cursor.

IL — Insert Line

```
Ec [ Pn L
```

Inserts `Pn` blank lines at the active cursor position. All the subsequent lines are moved down one line. The sequence is ignored when the cursor is outside the scrolling region.

ANSI Character Set Selection Sequences

The following character set control sequences are responded to by the terminal in ANSI mode:

	Mnemonic	Description
ANSI specified sequences	SCS	Select Character Set
	SGR	Select Graphic Rendition
HP private sequences	HPDHL	Make this line double height, double width
	HPDWL	Make this line double width
	HPSSL	Make this line standard size

SCS — Select Character Set

Ec (Ps — select base character set (G0)
Ec) Ps — select alternate character set (G1)

Where Ps can have the values:

Ps=B = configured language.

Ps=0 = ANSI line drawing set.

At power-on or after a hard reset, the default base character set (G0) is always the configured language set. The default alternate character set (G1) is always the ANSI line drawing set.

To invoke the base set (previously selected by the appropriate ANSI escape sequence) while operating in the alternate set, issue a Shift-in (SI) control character. The alternate set (G1) is enabled by the Shift-out (SO) control character. Unlike HP mode, the alternate set stays enabled until you perform a hard reset or send a Shift-in (SI) control character.

SGR — Select Graphic Rendition

`Ec [ Ps;...;Ps m`

Select video attribute(s) according to parameter value(s):

`Ps=0`	Turn off all attributes
`Ps=1`	Half-Bright
`Ps=4`	Underline
`Ps=5`	Blinking
`Ps=7`	Inverse Video

Attributes may be mixed in the same sequence. The attributes remain in effect until another SGR sequence is encountered.

HPDHL — Double Size Line

`Ec # Ps`

Sets the line containing the cursor to be one half of a double height double width line according to the parameter:

`Ps=3`	sets the line containing the cursor to become the top half of a double-height, double-width line.
`Ps=4`	sets the line containing the cursor to become the bottom half of a double-height, double-width line.

A double height double width line can contain up to 40 double-size characters.

The sequences must be used as a pair on adjacent lines.

The same character output (string) must be sent to both lines to form full double-height characters. Since there can be only 40 double-size characters per line.

For example, to set characters "THAN" as double size characters, use the sequences:

`Ec#3 THAN Cr Lf` →	THAN	top half of line
`Ec#4 THAN Cr Lf` →		bottom half of line

HPDWL — Double-Width Line

```
Ec # 6
```

Causes the line containing the cursor to become a double width single height line. A double-width line can contain up to 40 normal-size characters. If the line was single width, all characters to the right of the center of the screen (beyond the 40th) are not displayed but are retained in memory.

HPSSL — Single Size Line

```
Ec # 5
```

Causes the line containing the cursor to become a single width, single height line. If the line was one half of a double-height line, the other half of the double-height line is unaffected (and the corresponding half of the double-size character will remain on the display).

ANSI Terminal Status Sequences

In ANSI mode, the following status control sequences are responded to or sent by the terminal:

	Mnemonic	Description
ANSI specified sequences	DA	Device Attributes
	DSR	Device Status Request/Report
HP private sequences	HPID	Identity (same as DA)
	HPREQTPARM	Request Parameters
	HPREQTPARM	Report Parameters (sent in response to HPREQTPARM)

DA — Device Attributes (host to terminal)

```
Ec [ 0 c
```
or
```
Ec [ c
```

The host requests the terminal to send an identification sequence. The terminal will respond with:

```
Ec [ ? 1; 2c
```

DSR — Device Status Report (host to terminal and terminal to host)

```
Ec [ Ps n
```

Used in both host-to-terminal and terminal-to-host transfers to report the terminal status by means of the parameters. This sequence has four parameters:

■ CPU to terminal:	Ps=5	Return status report using DSR sequence.
	Ps=6	Return cursor position report using a CPR sequence.
■ Terminal to host:	Ps=0	Terminal ready, no malfunctions.
(response to DSR from host)	Ps=3	Malfunction detected, retry.

HPID — Identify

```
Ec Z
```

The HPID sequence has the same effect as the DA (Device Attributes) sequence.

HPREQTPARM — Request Terminal Parameters (host to terminal)

```
Ec [ Ps x
```

Sent by the host to request the terminal's operating parameters; where `Ps` can have the value `0` or `1`. Parameters are reported using a HPRPTPARM sequence (table B-7).

Table B-7. HPRPTPARM Report Parameters Mnemonics

PARAMETER MNEMONIC	PARAMETER VALUE	MEANING
`<par>`	1	No parity set
	2	Space parity
	3	Mark parity
	4	Parity set to ODD
	5	Parity set to EVEN
`<nbits>`	1	8 bits per character
	2	7 bits per character
`<xspeed>`	16	110 Bauds
(terminal	24	134.5 Baud
transmit speed)	32	150 Baud
and	48	300 Baud
`<rspeed`	56	600 Baud
(terminal	64	1200 Baud
receive speed)	72	1800 Baud
	88	2400 Baud
	72	4800 Baud
	112	9600 Baud
	120	19200 Baud

HPRPTPARM — Report Terminal Parameters (terminal to host)

```
Ec [ 3;<par>;<nbits>;<xspeed>;<rspeed>; 1 ; 0 x
```

Returned after a HPREQPARM sequence has been received to Report the terminal parameters. The sequence parameter mnemonics are listed in Table B-7. Note that on the 2393A, `<xspeed>` and `<rspeed>` always have the same value.

ANSI Terminal Control Sequences

	Mnemonic	Description
ANSI specified sequences	MC	Media Copy
	RIS	Reset to Initial State
HP private sequences	HPKPAM	Set Keypad to Application mode
	HPKPNM	Set Keypad to Numeric mode
	HPLF	Load flags

MC — MEDIA COPY

`Ec [ Ps i`

Controls the transfers between the terminal and the external device (provided the terminal is fitted with a second port) "Ps", as shown below:

`Ps=0` Copy All: Copy all data in display memory to printer, from line containing the cursor to end of memory.

`Ps=5` Turn on printer pass through mode. The terminal transmits received characters to the printer without displaying them on the screen. The terminal does not edit the data in any way. When in printer pass through mode, keyboard activity continues to be sent to the host.

`Ps=4` Turn off printer pass through mode.

`Ps=?5` Turn on Log Bottom mode: Incoming data is displayed on the screen, and when a line feed is encountered (either received from the host or as the result of a Wraparound), the line of data is sent to the printer.

`Ps=?4` Turn off Log Bottom mode: Incoming data is no longer sent to the printer as it is received.

RIS — Reset to Initial State

`Ec c`

Terminal performs a hard reset (equivalent to power-on). The terminal is then set as follows:

1. The cursor is displayed at the top left-hand corner of the screen (column 1, line 1), and the display memory is cleared.
2. The five labels associated with the Modes function are displayed at the bottom of the screen. Remote mode is on (asterisk in the label). The other four modes are off.
3. Certain keys and parameters are reset as follows:
   a. The keyboard is unlocked.
   b. The [Caps] key is reset (set to lower-case). The terminal may still generate upper-case characters if the Terminal Configuration menu's `Caps Lock` feature is `On`.
   c. The User Keys retain the definitions given to them before powering off.
   d. Cursor key and numeric pad application modes are reset.
   e. The scrolling region defauls to the entire screen.
   f. Tab stops, backspace, and multipage parameter as specified in the ANSI Configuration menu.
   g. The "insert character" edit function is switched OFF.
   h. The Record mode is switched OFF (this stops the terminal from automatically sending data to an external device).

HPKPAM — Keypad Application mode

`Ec =`

Sets the numeric keypad to Application mode, see Table B-3.

HPKPNM — Keypad Numeric mode

`Ec >`

Sets the numeric keypad to Numeric mode, see Table B-3.

HPLF — Load Flags (HP Private)

`Ec [ Ps;...q`

Controls the four programmable flags in the display Status Line as follows:

Ps Value	Effect on flags
0	Clear all flags
1	Illuminate flag "L1"
2	Illuminate flag "L2"
3	Illuminate flag "L3"
4	Illuminate flag "L4"

ANSI Terminal Mode Selection Sequences

During ANSI operations, various modes (or terminal controls) are available that operate in a similar manner to the straps in HP mode. These ANSI modes are listed in Table B-8. The modes labeled "set/reset" (in the value column) can be set and reset using the SM (Set mode) and RM (Reset mode) sequences, respectively. The individual modes are described below.

Table B-8 also lists other modes defined by ANSI Standard X3.64 1974. These modes, however, have a pre-defined fixed state that cannot be altered from the host computer or by the user. These modes are not affected by the SM and RM sequences.

Table B-8. ANSI And HP Private Modes

MNEMONIC	PARAMETER Ps	VALUE	FUNCTION	POWER-ON DEFAULT VALUE
			ANSI-specified Modes	
GATM	1	N/A	Guarded Area Transfer Mode	
KAM	2	Set/reset	Keyboard Action mode	Reset
CRM	3	Set/reset	Control Representation mode	Reset
IRM	4	Set/reset	Insert/Replace mode	Reset
SRTM	5	Reset	Status Report Transfer mode	
ERM	6	Set	Erasure mode	
VEM	7	N/A	Vertical Editing mode	
HEM	10	N/A	Horizontal Editing mode	
PUM	11	Reset	Positioning Unit mode	
SRM	12	Set	Send-Receive mode	
FEAM	13	Reset	Format Effector Action mode	
FETM	14	Reset	Format Effector Transfer mode	
MATM	15	N/A	Multiple Area Transfer mode	
TTM	16	N/A	Transfer Termination mode	
SATM	17	N/A	Selected Area Transfer mode	
TSM	18	Reset	Tabulation Stop mode	
EBM	19	Reset	Editing Boundary mode	
LNM	20	Set/reset	Line Feed/New Line mode	Reset
	38	Set/reset	Tek mode	Reset
			HP Private Modes	
HPCKM	? 1	Set/reset	Cursor Keys mode	Reset
HPANM	? 2	Set/reset	ANSI mode	Reset
HPCM	? 3	Set/reset	132 Column mode	Reset
HPSCLM	? 4	Set/reset	Scrolling mode	Reset
HPOM	? 6	Set/reset	Origin mode	Set
HPAWM	? 7	Set/reset	Autowrap mode	Reset
HPMPM	> 1	Set/reset	Multipage mode	Reset
HPMLM	> 2	Set/reset	Memory Lock mode	Reset

RM — Reset mode

`Ec [ Ps;...;Ps l`

Resets the terminal modes defined by the parameters Ps. The available parameters comprise all of the modes labeled "set/reset" in the "value" column in Table B-8 (these include ANSI and HP private modes). RM is the opposite of the SM (Set mode) sequence. These modes are described in detail below.

SM — Set mode

`Ec [ Ps;...;Ps h`

Sets the terminal modes defined by the parameter Ps. The available parameters comprise all of the modes labeled "set/reset" in the "value" column in Table B-8 (these include ANSI and HP private modes). SM is the opposite of the RM (Reset mode) sequence. These modes are described in detail below.

KAM — Keyboard Action mode

`Ec [ 2 h` (Set)

`Ec [ 2 l` (Reset) (Default state)

The set state disables the keyboard, and displays the message "`Kb Lockd`" in the screen Status Line.

In the reset state, the keyboard is enabled.

CRM — Control Representation mode

`Ec [ 3 h` (Set)

`Ec [ 3 l` (Reset) (Default state)

In the set state, control characters are displayed symbolically and the terminal does not execute the control function. A `CR` character, however, displays a `Cr` symbol and also causes a line feed and carriage return to be performed.

In the reset state, control characters are executed and not displayed.

The Control Representation mode can also be set/reset using the terminal's DISPLAY FUNCTIONS screen-labeled key.

IRM — Insert/Replace mode

`Ec [ 4 h` (Set)

`Ec [ 4 l` (Reset) (Default state)

Controls how entered or received characters affect characters previously displayed at the active cursor position.

In the reset state, each new displayable character overwrites the character previously displayed at the active cursor position. The cursor is then moved to the right (if it is not at the right margin), and no other character will be affected.

In the set state, each new character causes all characters on that line at or to the right of the active cursor position to be shifted one place to the right. The new character is inserted at the active position, and the cursor is then moved to the right. The screen status line displays the message "`Inschar`" when this function is active.

LNM — New Line mode

`Ec [ 20 h` (Set)

`Ec [ 20 l` (Reset) (Default state)

Controls the character(s) generated by the [Return] key.

In the set state, the characters "`Cr`" and "`Lf`" are generated.

In the reset state, only the character "`Cr`" is generated.

The LNM mode can also be set using the AUTO LF screen label.

Tek Mode

`Ec [ 38 h` (set)

`Ec [ 38 l` (reset) (Default state)

HPCKM — HP Cursor Keys mode.

`Ec [ ? 1 h` (set)

`Ec [ ? 1 l` (reset) (Default state)

In the reset state, the cursor keys send ANSI cursor movement control codes.

In the set state, the cursor keys send application functions control codes; see Table B-2.

HPANM — HP/ANSI/EM52 mode

`Ec [ ? 2 l` Leave ANSI mode and enter EM52 mode

`Ec <` Leave EM52 mode and enter ANSI mode

The terminal has the capability of operating with EM52 control sequences. When in ANSI mode, "EM52 mode" is entered with a standard ANSI control sequence (`Ec[?2l`); when in the EM52 mode, ANSI operation can be resumed with `ESC <`.

The other mode-changing sequences which will switch the terminal between HP mode and ANSI mode are:

`Ec & k 1 \` Leave HP mode and enter ANSI mode

`Ec & k 0 \` Leave ANSI mode and enter HP mode

Note Entering ANSI mode from HP mode causes the following to happen: Block mode is disabled, Memory Lock is turned off, the display is cleared, and the cursor is homed up.

HPCM — 132/80 Columns Mode

`Ec [ ? 3 h` (set)

`Ec [ ? 3 l` (reset) (Default state)

HPSCLM — HP Scrolling mode

`Ec [ ? 4 h` (Set)

`Ec [ ? 4 l` (reset) (Default state)

The reset state cause lines to jump when scrolling is used.

The set state causes lines to flow "smoothly" (at a maximum rate of six lines per second) when scrolling is used.

The HPSCLM mode can also be set/reset using the SMOOTH SCROLL screen label.

HPOM — HP Origin mode

`Ec [ ? 6 h` (set)

`Ec [ ? 6 l` (reset) (Default state)

In the reset state, the origin is the upper-left character position of the screen. Line and column numbers are therefore independent of top and bottom margins set using the HPSTBM sequence. The cursor may be positioned outside the margins with a Cursor Position (CUP) or Horizontal and Vertical Position (HVP) control sequence.

The set state causes the origin to be at the upper-left character position within the top and bottom margins set using the HPSTBM. The cursor may not be positioned outside the margins set by the HPSTBM sequence.

HPAWM — HP AutoWrap mode

`Ec [ ? 7 h` (set) (Default state)

`Ec [ ? 7 l` (reset)

In the reset state, automatic wraparound is not performed and when the cursor reaches the right margin it stays there (until explicitly moved) Consequently, characters received when the cursor is at the right margin overwrite any existing character.

In the set state, automatic wraparound is performed. Characters overflowing the right margin are written at the start of the next line, a scroll up is performed if required and if permitted.

HPMPM — HP Multipage mode

`Ec [ > 1 h` (Set)

`Ec [ > 1 l` (Reset) (Default state)

The set state selects multiple paging.

The reset state causes the terminal to become a single page terminal, with 24 lines of display memory.

When the terminal is either set or reset in this mode, the display is cleared and the cursor is homed up.

The HPMPM sequence temporarily overrides the `Multipage` parameter set in the ANSI Configuration menu. The terminal returns to the menu setting after power-on or a hard reset.

HPMLM — HP Memory Lock mode

`Ec [ > 2 h` (Set)

`Ec [ > 2 l` (Reset)

The set state turns on memory lock, which then functions similiar to the way it does in HP mode. The reset state turns Memory Lock off.

Memory Lock mode can also be reset using the terminal's MEMORY LOCK screen label.

If a scrolling region is defined, this sequence is ignored.

EM52 Control Sequence Summary

The following sub-sections describe the control sequences recognized and processed by the terminal when in EM52 mode.

Note that EM52 mode can only be entered from ANSI mode, using the ANSI HPANM control sequence (it is not possible to enter the EM52 mode directly from HP mode).

Cursor Up

```
Ec A
```

Moves the active cursor position upward by one line (without changing the column number). If an attempt is made to move the cursor past the top margin, it stops there (at the top margin).

Cursor Down

```
Ec B
```

Moves the active cursor position down by one line (without changing the column number). If an attempt is made to move the cursor past the bottom margin, it stops there (at the bottom margin).

Cursor Right

```
Ec C
```

Moves the active cursor position one column to the right. If an attempt is made to move the cursor past the right margin, it stops at the right margin.

Cursor Left

```
Ec D
```

Moves the active cursor position one column to the left. If an attempt is made to move the cursor past the left margin, it stops at the margin.

Select and Enable Alternate Character Set

```
Ec F
```

Select and Enable Base Character Set

```
Ec G
```

Cursor to Home

```
Ec H
```

Moves the cursor to the home-up position.

Reverse Line Feed

```
Ec I
```

Moves the active cursor position upward by one line (without changing the column number). If an attempt is made to move the cursor past the top margin, a roll down is performed (if required).

Erase to End Of Screen

```
Ec J
```

Erases all characters from the active cursor position to the end of the screen. The active cursor position is not changed.

Erase to End of Line

```
Ec K
```

Erases all characters from the active cursor position to the end of the line. The active cursor position is not changed.

Direct Cursor Address

```
Ec Y Pn Pn
```

Move the cursor to the specified position. This sequence has two parameters:

1. The first specifies the line number.
2. The second specifies the column number.

When specifying a line or column number, the parameter used must be sent as ASCII code of the value of the line/column number plus octal `37` (hexadecimal `1F`). For example: to specify line or column number "1", a space should be used (hex. `1F + 1`); to specify line or column "2" a "`!`" should be used (hex. `1F + 2`).

Identify

`Ec Z`

Causes the terminal to return its identifier sequence to the host CPU. The returned identifier sequence is: `Ec / Z`.

Enter Alternate Keypad mode

`Ec =`

Sets the numeric keypad to Application mode. (Refer to table B-3.)

Exit Alternate Keypad mode

`Ec >`

Sets the numeric keypad to Numeric mode. (Refer to table B-3.)

Enter ANSI mode

`Ec <`

Leave EM52 mode and enter ANSI mode.

Escape Sequence Summary

All ANSI and EM52 mode escape sequences are listed below:

ANSI Escape Sequences

MNEMONIC	SEQUENCE	DESCRIPTION
Cursor control		
CBT	Ec [Pn Z	Cursor Back Tab
CHA	Ec [Pn G	Cursor Horizontal Absolute
CNL	Ec [Pn E	Cursor Next Line
CPL	Ec [Pn F	Cursor Preceding Line
CPR	Ec [Pn; Pn R	Cursor position report
CUB	Ec [Pn D	Cursor Backward
CUD	Ec [Pn B	Cursor Down
CUF	Ec [Pn C	Cursor Forward
CUP	Ec [Pn; Pn H	Cursor Position
CUU	Ec [Pn A	Cursor up
HTS	Ec H	Horizontal Tab Set
HVP	Ec [Pn; Pn f	Horizontal and Vertical Position
IND	Ec D	Index
NEL	Ec E	Next line
RI	Ec M	Reverse Index
TBC	Ec [Ps g	Tab clear
VPA	Ec [Pn d	Vertical Position Absolute
VPR	Ec [Pn e	Vertical Position Relative
HPA	Ec [Pn `	Horizontal Position Absolute
HPR	Ec [Pn a	Horizontal Position Relative
HPHD	Ec [> 1 s	Home Down
HPHU	Ec [> 0 s	Home Up
HPSC	Ec 7	Save Cursor and attributes
HPRC	Ec 8	Restore Cursor and attributes

Display control

NP	Ec [Pn U	Next Page
PP	Ec [Pn V	Previous Page
SD	Ec [Pn T	Scroll Down
SU	Ec [Pn S	Scroll Up
	Ec [Pn <space> @	Scroll Left
	Ec [Pn <space> A	Scroll Right
HPSTBM	Ec [Pn; Pn r	Set top and bottom margins

Editing commands

DCH	Ec [Pn P	Delete Character
DL	Ec [Pn M	Delete line
ED	Ec [Ps J	Erase in display
EL	Ec [Ps K	Erase in line
IL	Ec [Ps L	Insert line

Character sets

SCS	Ec (Ps	Select base Character Set
	or Ec) Ps	Select alternate Character Set
SGR	Ec [Ps;...;Ps m	Select graphic rendition
HPDHL	Ec # 3 (Top half) Ec # 4 (Bottom half)	Make this line double-height, double-width
HPDWL	Ec # 6	Make this line double-width
HPSSL	Ec # 5	Make this line standard size

Terminal status

DA	Ec [0 c or Ec [c	Device Attributes
DSR	Ec [Ps n	Device status request/report
HPID	Ec Z	Same as DA (above)
HPREQTPARM	Ec [Ps x	Request Parameters
HPREPTPARM	Ec [Ps;..;Ps x	Report Parameters

Terminal control

MC	`Ec [ Ps i`	Media Copy
	`Ec [ ? Ps i`	Log Bottom
RIS	`Ec c`	Reset to initial state
HPKPAM	`Ec =`	Set Keypad to Application mode
HPKPNM	`Ec >`	Set Keypad to Numeric mode
HPLF	`Ec [Ps;...q`	Load flags

Terminal modes

RM	`Ec [ Ps;...;Ps l`	Reset mode(s)
SM	`Ec [ Ps;...;Ps h`	Set mode(s)

MODE NAME	TO SET		TO RESET	
Cursor Keys mode	Application	`Ec [ ? 1 h`	Cursor	`Ec [ ? 1 l`
Keypad mode	Application	`Ec =`	Numeric	`Ec >`
ANSI/HP mode	HP to ANSI	`Ec & k 1 <`	ANSI to HP	`Ec & k 0 \`
ANSI/EM52 mode	EM52 to ANSI	`Ec <`	ANSI to EM52	`Ec [ ? 2 l`
Column mode	80 columns	`Ec [ ? 3 h`	132 columns	`Ec [ ? 3 l`
Scrolling mode	Smooth	`Ec [ ? 4 h`	Jump	`Ec [ ? 4 l`
Origin mode	Relative	`Ec [ ? 6 h`	Absolute	`Ec [ ? 6 l`
Wraparound	On	`Ec [ ? 7 h`	Off	`Ec [ ? 7 l`
Keyboard Action	Kbd disabled	`Ec [ 2 h`	Kbd enabled	`Ec [ 2 l`
Control repres.	Display fnctns	`Ec [ 3 h`	Exec fnctns	`Ec [ 3 l`
Insert/Replace mode	Insert	`Ec [ 4 h`	Replace	`Ec [ 4 l`
Line feed/new line	New line	`Ec [ 20 h`	Linefeed	`Ec [ 20 l`
Tek/normal mode	Tek mode	`Ec [ 38 h`	Normal	`Ec [ 38 l`
Multi-page mode	Multipage	`Ec [ > 1 h`	Single page	`Ec [ > 1 l`
Memory Lock	Locked	`Ec [ > 2 h`	Unlocked	`Ec [ > 2 l`

EM52 Escape Sequences

Ec A	Cursor up
Ec B	Cursor down
Ec C	Cursor Right
Ec D	Cursor Left
Ec F	Enter graphics mode
Ec G	Exit graphics mode
Ec H	Home cursor
Ec I	Reverse linefeed
Ec J	Erase to end of screen
Ec K	Erase to end of line
Ec Y Ln Cn	Direct cursor address
Ec Z	Identify
Ec =	Enter Alternate Keypad mode
Ec >	Exit Alternate Keypad mode
Ec <	Go to ANSI mode

C Error Messages

Introduction

When an error occurs, the terminal displays an error message at the bottom of the screen. Some errors indicate wrong keyboard input. Others indicate improper configuration settings. A few signal a terminal malfunction.

Error Messages

The terminal generates several kinds of status checks and diagnostic error messages. You should concern yourself with two: user-error messages and system error messages.

Most user error messages occur when you enter data that the terminal was not expecting or request a service that the terminal cannot perform. However, some errors result from incompatible settings in the configuration menus.

Error messages appear on lines 25 and 26, replacing the function key labels. Pressing the [Return] key clears the error message, restores the labels, and unlocks the keyboard.

Table C-1 lists the error messages and their meanings.

Table C-1. Error Messages

MESSAGE	MEANING
`DC ERROR #`	All datacomm and manufacturing self-tests must be done with a datacomm test hood attached to each of the datacomm ports. If a test hood is not attached, a `DC ERROR 26` will always occur. If a test hood is attached and an error occurs, write down the error and report it to your HP service engineer. where # is: `13` Control line error. Caused by bad test hood or faulty hardware. The `ASSERTED=`, `EXPECTED=`, and `RECEIVED=` indicate which control lines are bad. `20` Bad character. The data looped back was bad. `21` Framing error. The character was returned with an incorrect number of bits. `22` Overrun error. The character returned was overrun. `23` Parity error. The character returned had the wrong parity. `24` Fast baud rate. The baud rate generator is running too fast. `25` Slow baud rate. The baud rate generator is running too slow. `26` No character returned. This is most often caused by testing the datacomm with no test hood installed.

Table C-1. Error Messages (continued)

MESSAGE	MEANING
`Default Configurations Used`	If during power up, the nonvolatile RAM was found to be in error, the terminal will use its default configuration settings. This message is displayed to warn you that the default configuration selections may be different from those you saved in nonvolatile memory.
`Device Busy`	This error indicates that you tried to use an HP-IB or other device that was already busy.
`ERROR: HP-HIL loop not connected !`	Your keyboard loop has been disconnected. Reconnect any loose HP-HIL cables back in. You should use a soft reset [Shift] [Reset] to assure that the terminal recognizes all of your HP-HIL devices after re-attaching them.
`Functions Locked`	The System, Menu, configuration, and some modes keys have been locked by the currently running application using the "`Ec&jS`" sequence. This can be cleared by a soft reset, [Shift] [Reset].
`Illegal Unit Device`	An illegal HP-IB unit device number was selected by the `Ec&p` escape sequence.
`Illegal for edit type: ALPHABETIC`	You tried to enter a non-alphabetic character into a field that was defined to accept alphabetic characters.

Table C-1. Error Messages (continued)

MESSAGE	MEANING
Illegal for edit type: NUMERIC	You tried to enter a non-numeric character into a field that was defined to accept numeric characters only.
Illegal or No Destination Device	You, or a program, tried to send data to a printer or plotter when no "to device" was selected. Go to the "to devices" function keys and select the appropriate device.
Illegal or No Source Device	An application program tried to copy data without selecting a source device, or it selected an illegal one. Correct the program.
Invalid Configuration	You tried to save a configuration which contained a something that didn't make sense. For example, Parity/Databits = NONE/8 and Check Parity = Yes (the terminal can't check parity if it is set to NONE).
MEMORY FULL	The video memory has run out. Either memory lock is on and you have filled it all, or you or an application was trying to add another field and there wasn't enough memory left to do so.
Pod/Driver Types Not Matched	An optional interface module was added while the power was on. You must turn the terminal off and then back on, or hard reset the terminal (Shift Ctrl Reset) to correct this. Any data on the screen will be lost.

Table C-1. Error Messages (continued)

MESSAGE	MEANING
`Softkey definitions too long to save`	There is not enough memory left to save the function keys. Shorten them and try again.
`Source = Destination`	You or an application tried to copy data from a device to itself. Select a new destination device in the "to devices" softkeys.
`Use NEXT or PREVIOUS Key`	You tried to type a character into a configuration field that can be changed only by using the NEXT or PREVIOUS function keys.
`Value Out of Range`	You typed a value into a configuration field that is too large or too small. Try a new value. Check your reference manual for allowable values.

D Keyboards and Character Sets

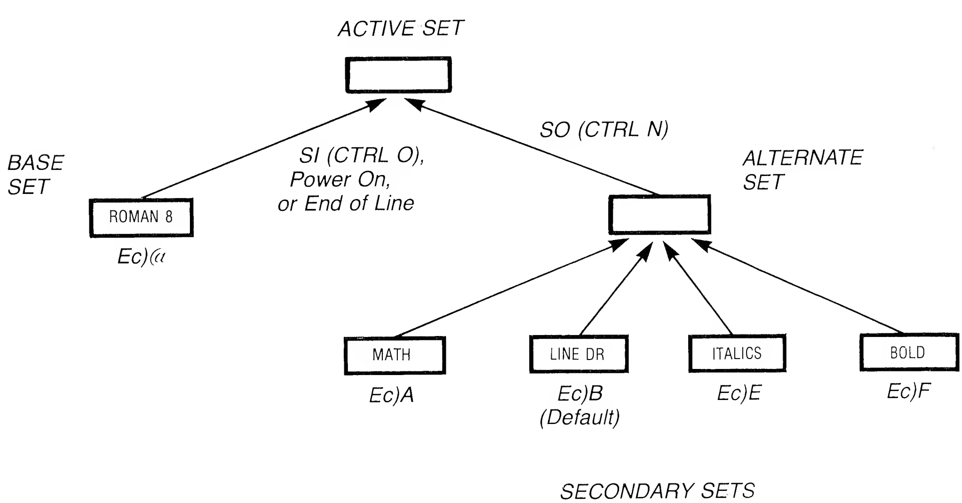
Introduction

This section is concerned with character sets, languages, and keyboards supported by the terminal.

Character Sets

The terminal has a base character set and four secondary character sets. Any one of the four secondary sets can be selected as the alternate character set. The active character set is selected from either the base set or the alternate set (figure D-1). (The active character set is the set from which characters are selected for display, whether the source is the keyboard or the datacomm line.)

Figure D-1. Character Set Selection



Base Character Set

The base character set is the Roman 8 set, which, except for the four secondary sets, contains all the characters displayed or recognized by the terminal. It consists of two subsets: the USASCII set and the Roman Extension set (table D-1). (All Roman 8 characters are listed in table D-5, at the end of this section.)

Table D-1. Roman 8 Character Set

b4	b3	b2	b1																	
				b8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				b7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
					0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p				¯	â	Å	Á	Þ
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q			À		ê	î	Ã	þ
0	0	1	0	2	STX	DC2	″	2	B	R	b	r			Â		ô	Ø	ã	
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s			È	°	û	Æ	Ð	
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t			Ê	Ç	á	å	đ	
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u			Ë	ç	é	í	Í	
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v			Î	Ñ	ó	ø	Ì	–
0	1	1	1	7	BEL	ETB	’	7	G	W	g	w			Ï	ñ	ú	æ	Ó	¼
1	0	0	0	8	BS	CAN	(	8	H	X	h	x			´	¡	à	Ä	Ò	½
1	0	0	1	9	HT	EM	)	9	I	Y	i	y			`	¿	è	ì	Õ	ª
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z			ˆ	¤	ò	Ö	õ	º
1	0	1	1	11	VT	ESC	+	;	K	[	k	{			¨	£	ù	Ü	Š	«
1	1	0	0	12	FF	FS	,	<	L	\	l	|			˜	Y	ä	É	š	■
1	1	0	1	13	CR	GS	-	=	M	]	m	}			Ù	§	ë	ï	Ú	»
1	1	1	0	14	SO	RS	.	>	N	^	n	~			Û	ƒ	ö	ß	Ÿ	±
1	1	1	1	15	SI	US	/	?	O	_	o	DEL			₤	¢	ü	Ô	ÿ	

USASCII Set (columns 0–7) — Roman Extension Set (columns 10–15)

USASCII Character Set. The USASCII character set consists of the characters with ASCII decimal codes 0 through 127. Character codes 0 through 31 are control codes, used for control of data and datacomm operations. These codes can be displayed only in Display Functions mode.

The remaining codes (32–127) are displayable characters, consisting of a space, the numbers 0–9, the upper and lower case letters of the English alphabet, and punctuation marks.

Roman Extension Character Set. The Roman Extension set is composed largely of special characters used in non-USASCII national languages, including associated diacritic marks. These characters are assigned decimal codes ranging from 161–254. Table D-2 lists these characters and the keyboard with which each is associated.

In table D-2, no correlation necessarily exists between the decimal value for a key and the corresponding position of the key on a keyboard. For example, the decimal value "92" maps to the back slant (\) on the USASCII keyboard. On the French keyboards, "92" maps to c-cedilla (ç). However, the c-cedilla key on the French keyboard does not physically correspond to the USASCII keyboard's back slant key.

Table D-2. Special National Language Characters

	KEYBOARD	DECIMAL VALUE											
LANGUAGE	OPTION #	35	39	64	91	92	93	94	96	123	124	125	126
USASCII	(standard)	#	’	@	[	\	]	^	‘	{	|	}	~
Swedish	101	#	’	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Norwegian	102	#	´	@	Æ	Ø	Å	^	`	æ	ø	å	¨
French	103	£	’	à	°	ç	§	^	`	é	ù	è	¨
German	104	£	´	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
United Kingdom	105	£	’	@	[	\	]	^	`	{	|	}	~
European Spanish	106	#	´	@	¡	Ñ	¿	°	`	{	ñ	}	¨
French Canadian	107	#	’	@	[	ç	]	^	`	é	Ç	É	¨
English Canadian	108	#	’	@	[	ç	]	^	`	é	Ç	É	¨
Italian	109	£	’	§	°	ç	é	^	ù	à	ò	è	ì
Dutch	110	#	´	@	ç	\	§	^	`	ƒ	|	¨	~
Finnish	111	#	’	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Danish	112	§	´	@	Æ	Ø	Å	^	`	æ	ø	å	¨
German Swiss	113	£	´	à	°	ç	§	^	`	ä	ö	ü	¨
French Swiss	114	£	´	à	°	ç	§	^	`	ä	ö	ü	¨
Latin American Spanish	115	#	´	@	¡	Ñ	¿	^	`	’	ñ	ç	¨
Belgian	116	£	’	á	°	ç	§	^	`	é	ù	è	¨

Secondary Character Sets

In addition to the base character set, the terminal has four secondary character sets. The secondary sets are: Math, Line-Drawing, Italics, and Bold characters. Figures D-2 and D-3 illustrate the Math and Line Drawing character sets.

Figure D-2. Math Character Set

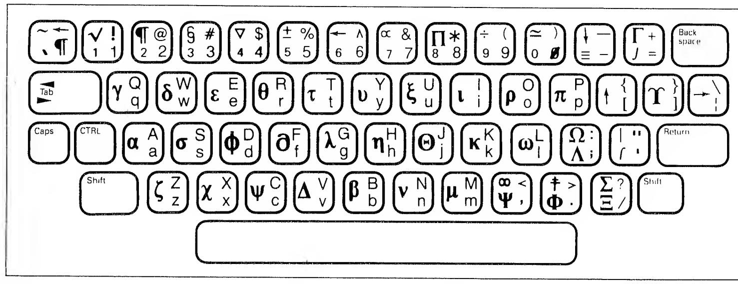
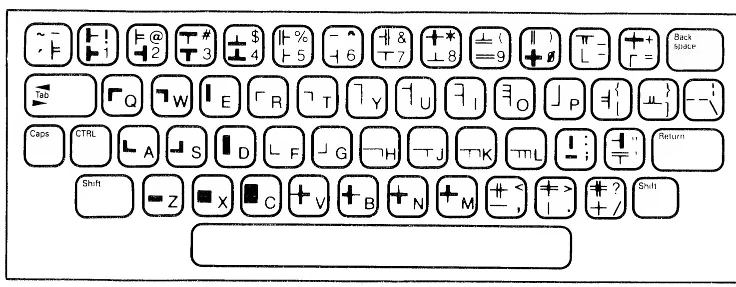


Figure D-3. Line Drawing Character Set



Accessing the Base Character Set

At power on and at the end of a line, during data entry, the base character set automatically becomes the active character set.

At any other time, The base set can be made the active set by a "shift in" operation. From the keyboard, this is done by pressing the [CTRL] and [N] keys together. From a program, an "SI" character (ASCII decimal code 15) is sent to the terminal.

Accessing a Secondary Character Set

Accessing any one of the secondary character sets is a two step operation. First, you select it as the alternate character set, then you activate the alternate character set.

If you select the base set as the alternate set, there will be no distinction between characters displayed when the alternate character set is enabled and when it is not.

Once the alternate character set is activated, all non-control characters received from the keyboard or over datacomm lines are displayed as characters from the alternate set until one of the following conditions occurs:

- The end of the line is reached.
- The Base set is "shifted-in" ([CTRL] and [O], pressed together).
- A display enhancement (underline, inverse video, blinking, or half-bright) is encountered.

Selecting the Alternate Character Set. A secondary character set is selected as the alternate set by entering it in the Alternate Set field of the Terminal Configuration menu. This can be done either from the keyboard or a program.

From the Keyboard. From the keyboard, you select one of the secondary sets as the alternate character set by pressing the [System] key, then **enhance video**, then **etc.** to display the following function key labels:

f1	f2	f3	f4	f5	f6	f7	f8
CHANGE TO BASE	CHANGE TO MATH	CHANGE TO LINE	CHANGE TO BOLD	CHANGE TO ITALC			etc.

When the alternate character set is selected using the function keys, the selection on the Terminal Configuration menu is changed automatically.

From a Program. From a program, you select one of the secondary sets as the alternate set with one of the following escape sequences:

CHARACTER SET	ESCAPE SEQUENCE
Base	ESC) @
Math	ESC) A
Line Drawing	ESC) B
Italics	ESC) E
Bold	ESC) F

Activating the Alternate Character Set. You access the characters of the selected alternate character set by executing a "shift-out" operation.

From the Keyboard. From the keyboard, you shift-out from the base set by simultaneously pressing the [CTRL] and [N] keys.

From a Program. An executing program can shift-out from the base set by issuing an ASCII <SO> code (ASCII decimal code 14).

Languages

The following paragraphs discuss selection of the language(s) to be used, the effect of 7-bit and 8-bit modes on data in datacomm operations, and use of the Roman Extension set of characters from the keyboard.

Language Selection

The terminal might use two languages at any given time. The language used for the function key labels, the status line, and error messages is selected in the "`Language`" field of the Global Configuration menu. The values for this field are:

```
English     Italiana
Dansk       Nederlands
Deutsch     Norsk
Espanol     Suomi
Francais    Svensk
```

The keyboard language and the datacomm line language is selected automatically, by the terminal firmware, for compatibility with the keyboard connected to the terminal.

Datacomm Operations

The terminal has two modes of operation that affect how characters received from datacomm are interpreted by the terminal. The modes are named for the number of significant bits they contain. In 8-bit mode all bits are significant; thus no bit is available for parity checking. In 7-bit mode, the seven low-order bits contain valid data. The eighth bit may be used for parity checking, or it may be ignored.

7-Bit Mode. When the terminal is configured for 7-bit mode, the least significant seven bits of the character byte determine the character's identity. That is, the seven bits are translated into the appropriate character, according to the language selected by the terminal firmware to be compatible with the connected keyboard. Seven bits limits the number of useable characters to 128.

Note

In 7-bit mode, the only accessible alphanumeric characters are those available from the language associated with the connected keyboard. For example, while in 7-bit mode, if the current keyboard is English, the host computer and terminal can only recognize the standard ASCII characters.

The special characters used in national languages, as listed in table D-2, are assigned decimal codes in the range 161–254. Since character access is limited to the range 0–127 in 7-bit mode, the special characters would be inaccessible without special attention. To access the special characters, their decimal codes are mapped to new decimal codes, in 7-bit mode, as listed in table D-2.

Example

Refer to table D-2. Notice that, when the terminal is in 7-bit mode, if the host sends the decimal value 35 and either a USASCII, Swedish, Norwegian, Spanish, Latin American Spanish, French-Canadian, Canadian-English, Dutch, or Finnish keyboard is connected to the terminal, the terminal interprets the characters as the number sign (#).

If the host sends the same code, however, to a terminal with either a French, German, Italian, United Kingdom, Swiss German, Swiss French, Danish, or Belgian keyboard attached, the terminal interprets the code as "£".

To configure the terminal for 7-bit mode, set the `ASCII 8 Bits` field in the Terminal Configuration menu to "`No`" or send the escape sequence:

```
Ec&k0I
```

Note You or the host computer may always access the characters in the Line Drawing or Math set if the set is chosen as the alternate character set. This is true regardless of the language chosen or the bit mode used.

8-Bit Mode. With the terminal configured for 8-bit mode, all eight bits are available for addressing so that the host and terminal can access any alphanumeric character in the

range 0-255. This enables direct access to any character in the Roman 8 set. No decimal code mapping is necessary, as in 7-bit mode.

8-bit mode also changes the operation of the graphics/ numeric keypad. When the terminal is in 8-bit mode and the keypad is defined for numeric operation, the numeric keys have shifted functions as shown in table D-3. This is to allow access to commonly used programming symbols from any language.

Table D-3. Shifted Functions of the Numeric Pad in 8-Bit Mode

KEY	SHIFTED CHARACTER
0	^
.	~
1	{
2	
3	}
4	[
5	\
6	]
7	#
8	'
9	@
-	transforms pad to Graphics operation

To configure the terminal for 8-bit mode, set the `Parity/DataBits` field on the datacomm configuration menu to "`None/8`". Also, set the `ASCII 8 Bits field` on the Terminal Configuration menu to "`Yes`".

The `Parity/DataBits` field must be set manually, but the `ASCII 8 Bits` field can be set to "`Yes`" with the following escape sequence:

```
Ec&k 1I
```

When using 8-bit mode, you must set the `Parity` field in the Datacomm Configuration menu to "`None`". Failure to do so will cause data communication problems with the host computer.

Accessing Any Character from the Keyboard

You can access any character of any supported language by two methods: using the [Extend char] key or using the [CTRL] key and the ASCII decimal code of the desired character.

Diacritic Marks. Certain Roman Extension characters contain diacritic marks, such as umlauts, tildes, and graves. These characters, unless they are present on the keyboard, require special handling for display on the screen.

Upon entering a diacritic mark (such as ^ or '), the cursor remains in the same position. (The diacritic mark is said to be "mute" if the cursor doesn't move when the diacritic is typed in.) If the next-typed character can be combined with that mark, the two characters are merged before the cursor advances to the next position. (The acceptable characters form the set: a, e, i, o, u, n, y, A, E, I, O, U, N, Y.) If the next-typed character is unacceptable, the character just entered replaces the diacritic mark as the displayed character and the cursor advances to the next position.

The case may arise when you want to enter just the diacritic mark. In such a case, you type a space after the diacritic character. The diacritic character remains displayed and the cursor advances to the next character position.

Extended Characters Mode. Extended Characters mode grants full access to the Roman Extension character set. Using this mode, you may select any character foreign to the selected language. Two critical points are:

1. You must configure the terminal for Extended Characters mode (set the `ASCII 8 Bits` field on the Terminal Configuration menu to "`Yes`"). With this condition set, an "`Ext Char`" message will appear on the status line when you press the [Extend char] key.

2. You can only enter Extended Characters mode through the keyboard.

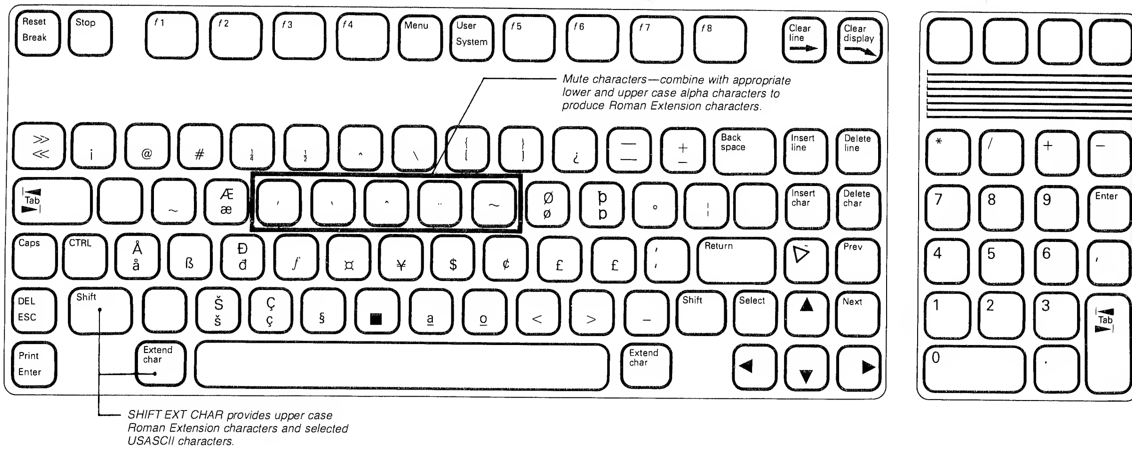
Table D-4 shows the Roman Extension characters which replace the keyboard characters in Extended Characters mode. Figure D-4 shows the correspondence between key caps on the USASCII keyboard and the Roman Extension characters they generate.

Table D-4. Roman Extension Characters Which Replace Keyboard Characters in Extended Characters Mode

KEYBOARDS	CHARACTERS													
Decimal Code	35	39	60	62	64	91	92	93	94	96	123	124	125	126
USASCII	#	’	<	>	@	[	\	]	^	‘	{	|	}	~
Belgian	£	’	é	è	à	°	ç	§	^	‘	é	ù	è	¨
Danish	§	´[2]	<	>	@	Æ	Ø	Å	^	`[2]	æ	ø	å	¨[2]
Dutch	#	’	<	>	@	ç	\	§	^[2]	`[2]	ƒ	|	´[2]	¨[2]
Finnish	#	’	<	>	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
French	£	’	<	>	à	°	ç	§	^[1]	‘	é	ù	è	¨[1]
English Canadian and French Canadian	#	’	<	>	@	[	ç	]	^[2]	`[2]	é	Ç	É	¨[2]
French Swiss and German Swiss	£	´[2]	é	è	à	°	ç	§	^[2]	`[2]	ä	ö	ü	¨[2]
German	£	’	<	>	§	Ä	Ö	Ü	^	‘	ä	ö	ü	ß
Italian	£	’	<	>	§	°	ç	é	^[2]	ù	à	ò	è	ì
Norwegian	#	´[2]	<	>	@	Æ	Ø	Å	^	`[2]	æ	ø	å	¨[2]
European Spanish	#	´[1]	<	>	@	¡	Ñ	¿	°	`[1]	’	ñ	ç	¨[1]
Latin American Spanish	#	´[1]	<	>	@	¡	Ñ	¿	^	`	’	ñ	ç	¨[1]
Swedish	#	’	<	>	É	Ä	Ö	Å	Ü	é	ä	ö	å	¨
United Kingdom	£	’	<	>	@	[	\	]	^	‘	{	|	}	~

Notes: [1]This diacritic is mute in both 7-bit and 8-bit modes.
[2]This diacritic is mute in 8-bit mode only.

Figure D-4. Roman Extension Characters Accessed Using the "Extend char" Key



Being a keyboard function, Extended Characters mode only affects data entered from the keyboard. It has no effect on data received over the datacomm lines. However, the following special circumstances exist:

1. If you enter any control codes (simultaneously pressing the [CTRL] key and another appropriate key), the code is interpreted as if Extended Characters mode were off.
2. The diacritic marks for the language associated with the connected keyboard can be entered directly from the keyboard, or entered with the [Extend char] key.

Using Extended Characters Mode. To display a character from the Roman Extension character set, press and hold down the [Extend char] key and at the same time press another key. For example, press [Extend char] and [e] together; the "æ" character will be displayed.

When the [Extend char] key is held down, an "Ext Char" entry appears on the status line at the bottom of the screen.

Some keys access a different Roman Extension character when pressed with the [Shift] key. As an example, press [Shift], [Extend char], and [e] at the same time to display the "Æ" character (which is the shifted Roman Extension character associated with the [e] key). A key associated with only one Roman Extension character displays that character whether you press the key with [Shift] [Extend char], or just with [Extend char].

Certain keys access special Roman Extension characters called "diacritic marks". Refer to "Diacritic Marks", earlier in this section, for detailed information.

The [q], [z], and [\] keys have no Roman Extension characters associated with them. If you type one of these keys while pressing the [Extend char] key, nothing is displayed and the cursor remains stationary.

Exiting Extended Characters Mode. To leave Extended Characters mode, simply release the [Extend char] key. This returns the keyboard to normal operation, and subsequent keystrokes produce normal characters.

CTRL Key Access to Any Character. You can access any character from the Roman Extension set by using [CTRL] followed by the ASCII decimal equivalent for the desired character. For example, [CTRL], [2], [3], [8] displays a "Ÿ".

National Language Keyboards

17 national language keyboards are supported for the terminal. Keyboards for the following countries are available: United States, Belgium, Canadian English, Canadian French, Danish, Dutch, Finnish, French, German, Italian, Norwegian, Spanish (Europe), Spanish (Latin America), Swedish, Swiss French, Swiss German, and the United Kingdom (figures D-5 through D-21).

Figure D-5. United States (USASCII) Keyboard

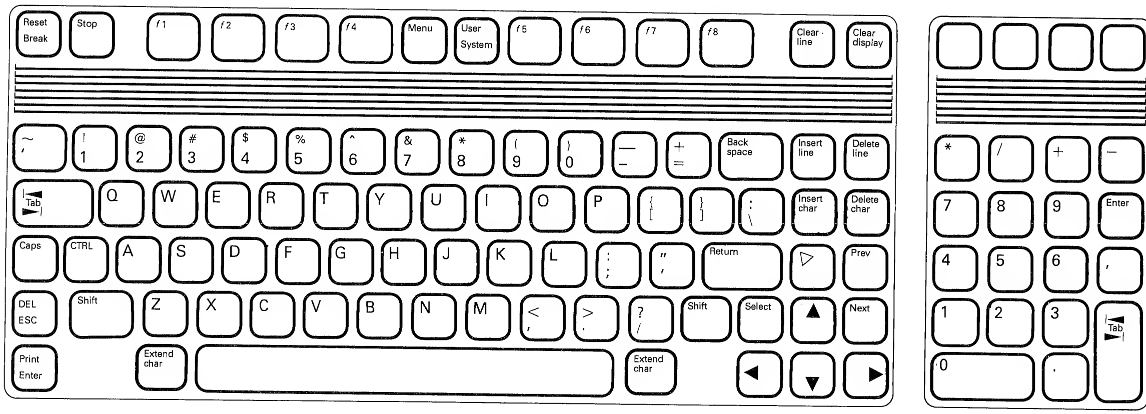


Figure D-6. Belgian (Flemish) Keyboard

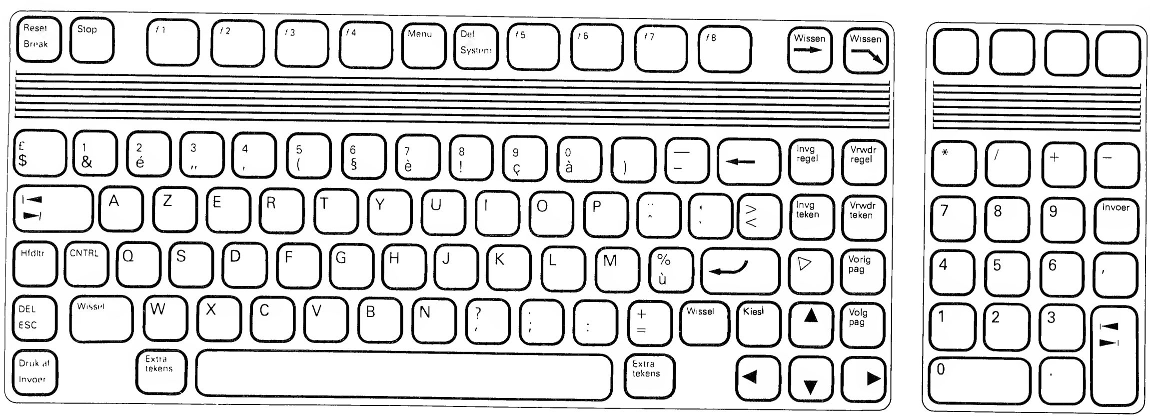


Figure D-7. Canadian English Keyboard

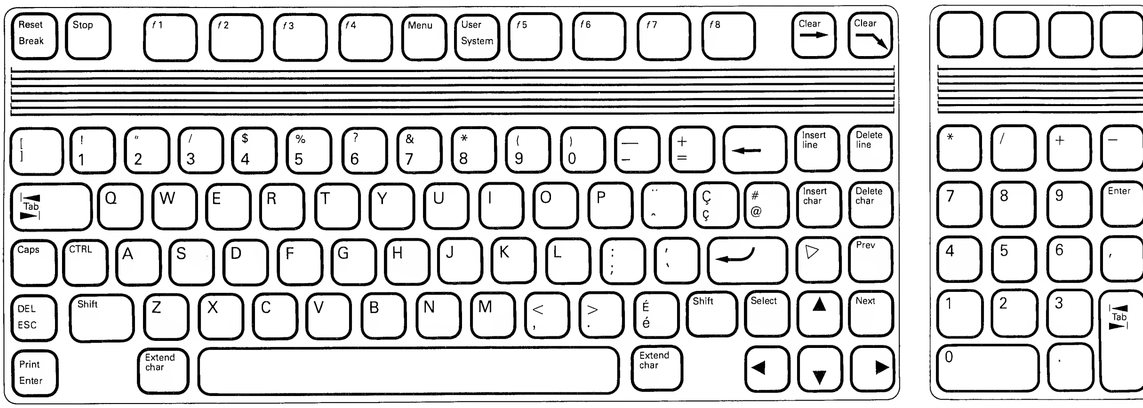


Figure D-8. Canadian French Keyboard

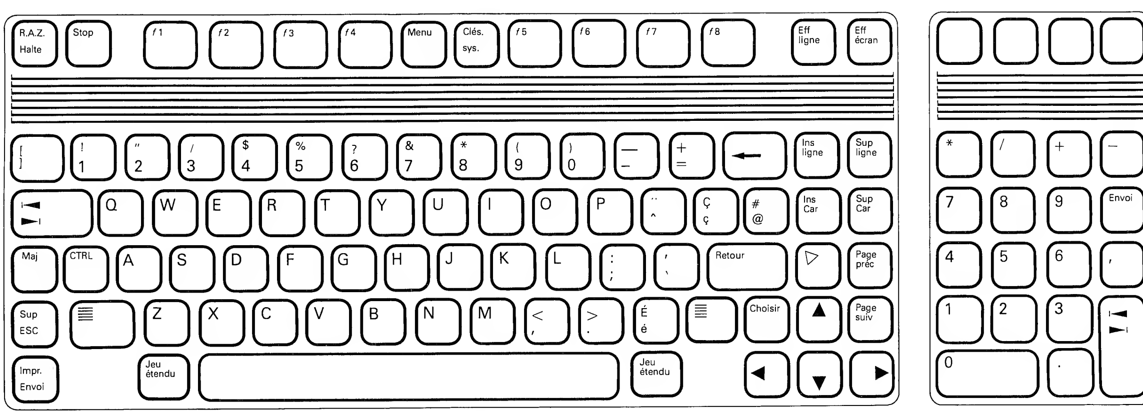


Figure D-9. Danish Keyboard

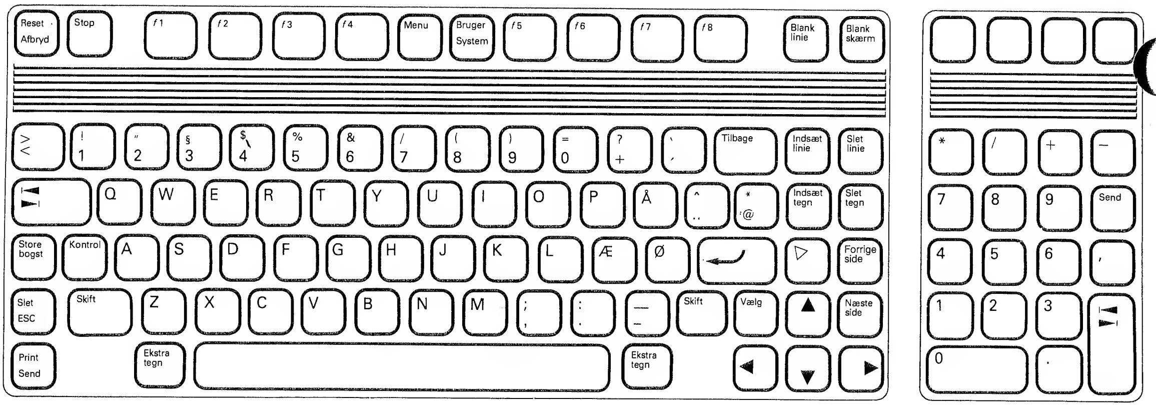


Figure D-10. Dutch Keyboard

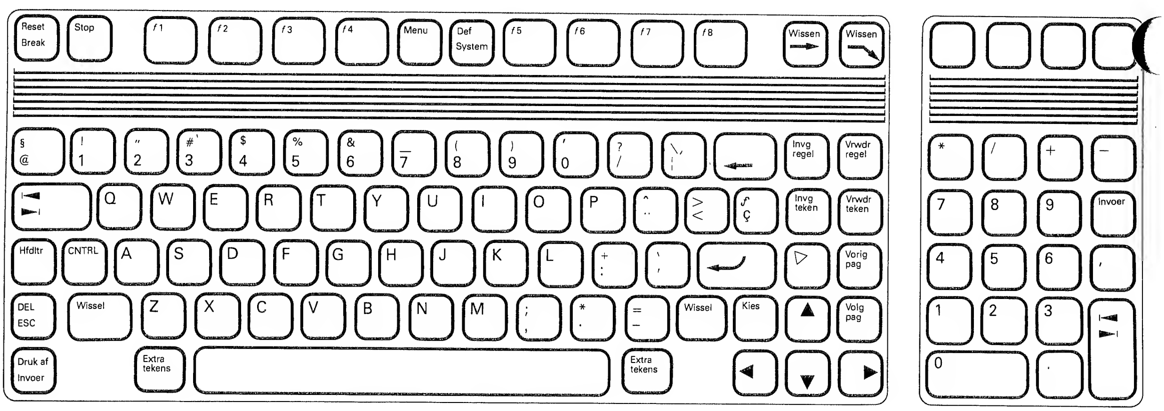


Figure D-11. Finnish Keyboard

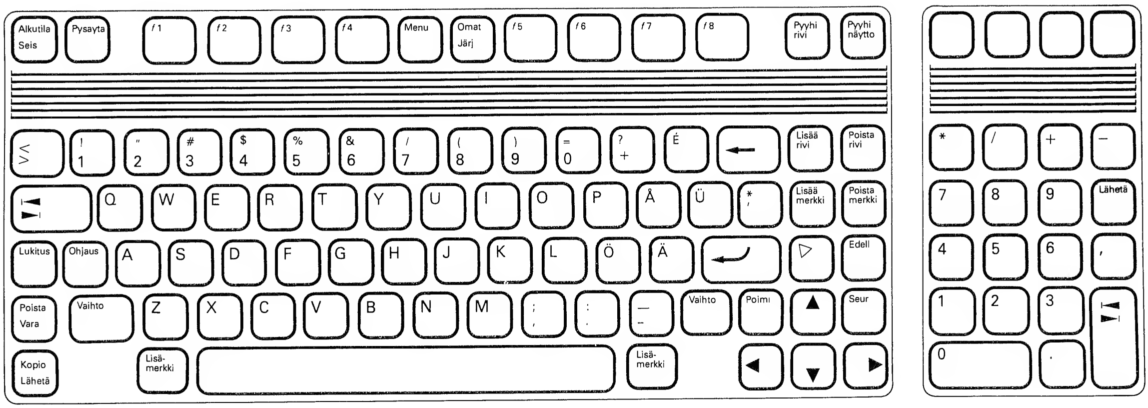


Figure D-12. French Keyboard

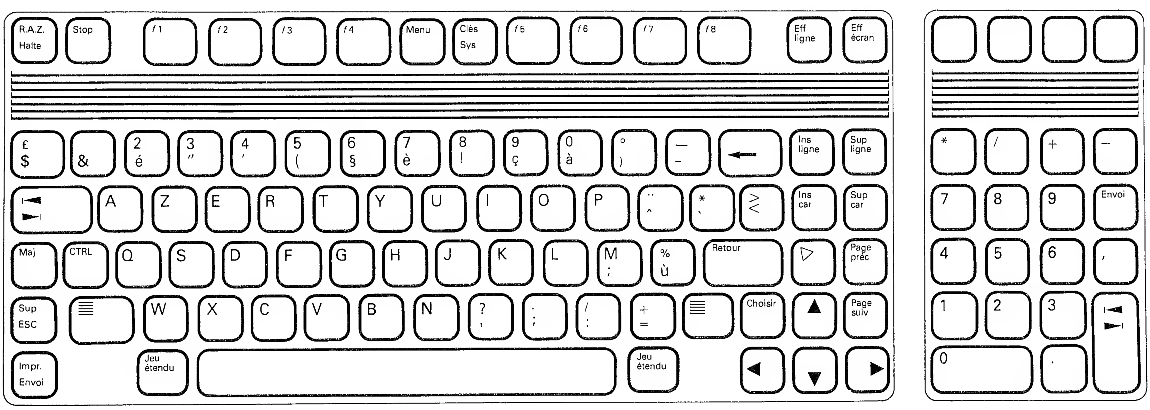


Figure D-13. German Keyboard

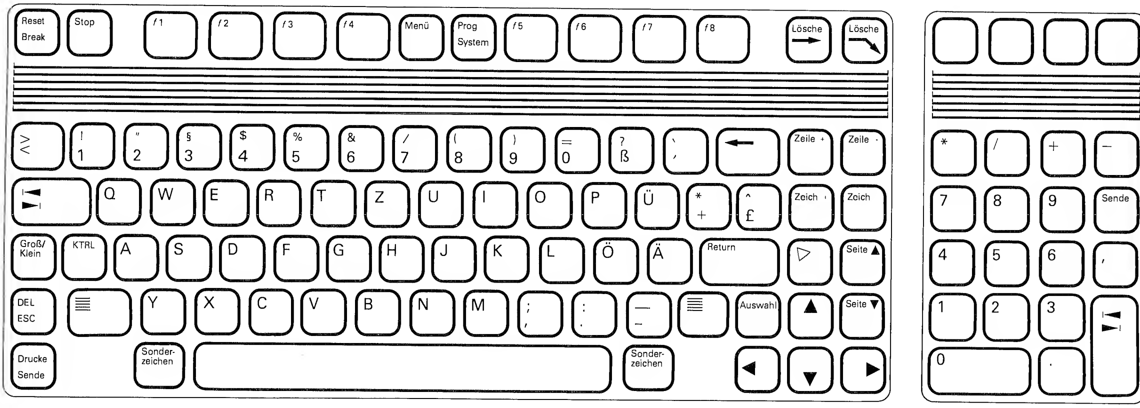


Figure D-14. Italian Keyboard

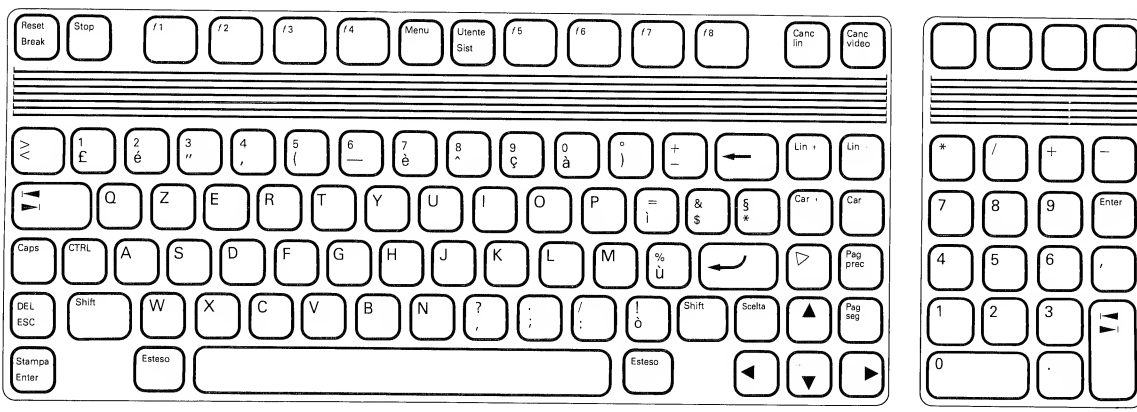


Figure D-15. Norwegian Keyboard

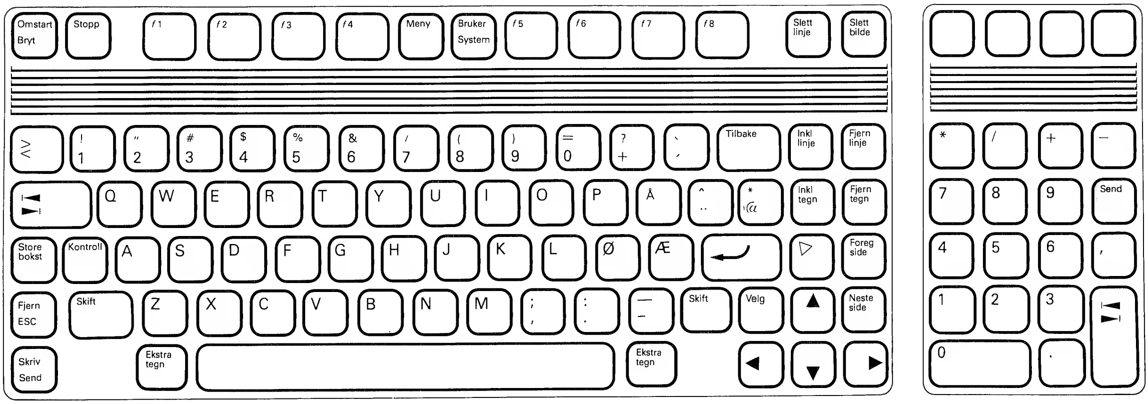


Figure D-16. Spanish (European) Keyboard

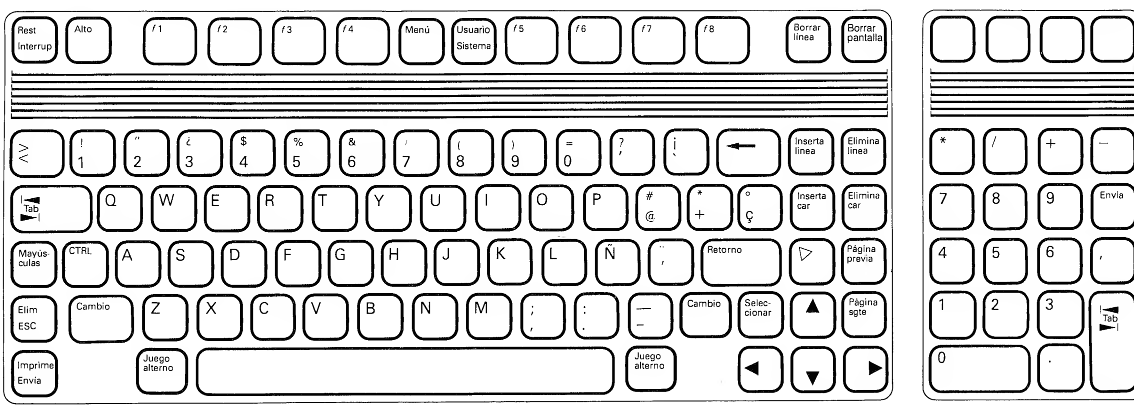


Figure D-17. Spanish (Latin American) Keyboard

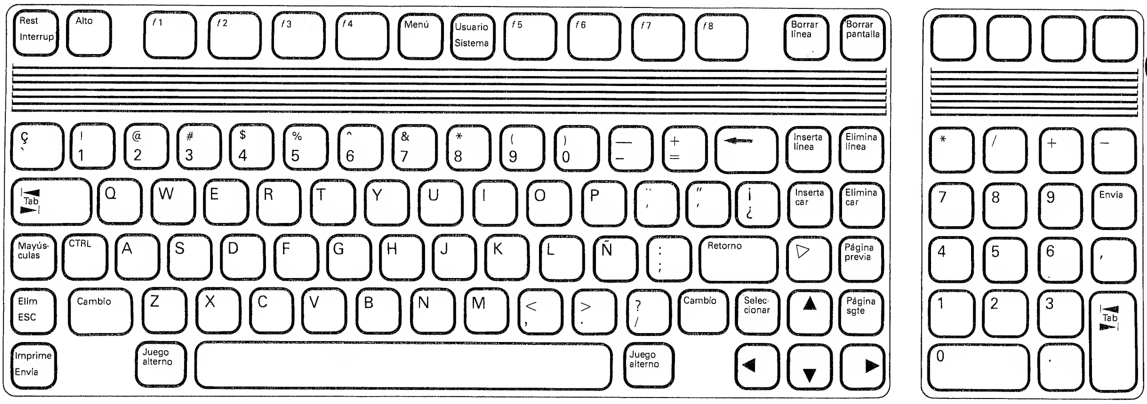


Figure D-18. Swedish Keyboard

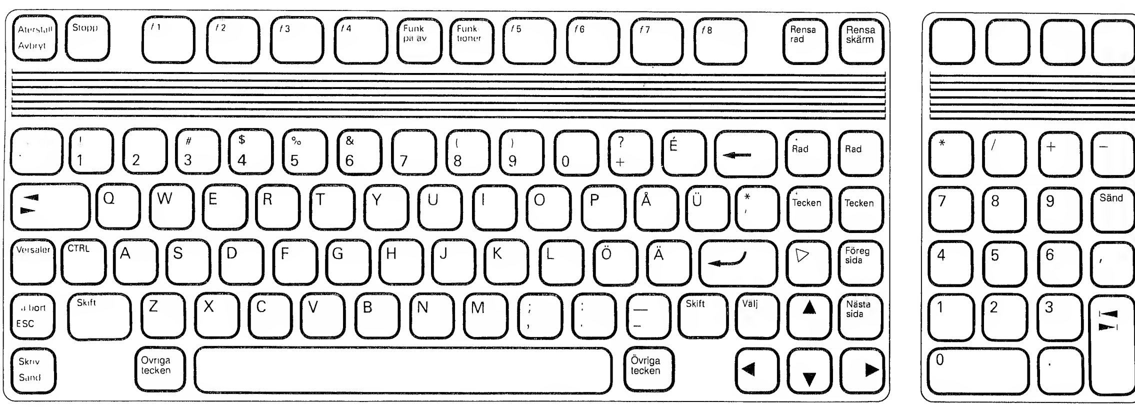


Figure D-19. Swiss French Keyboard

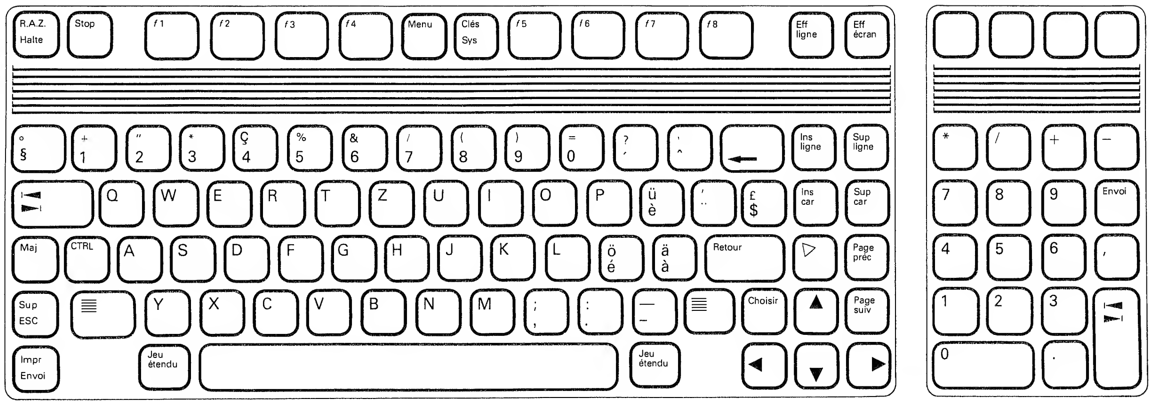


Figure D-20. Swiss German Keyboard

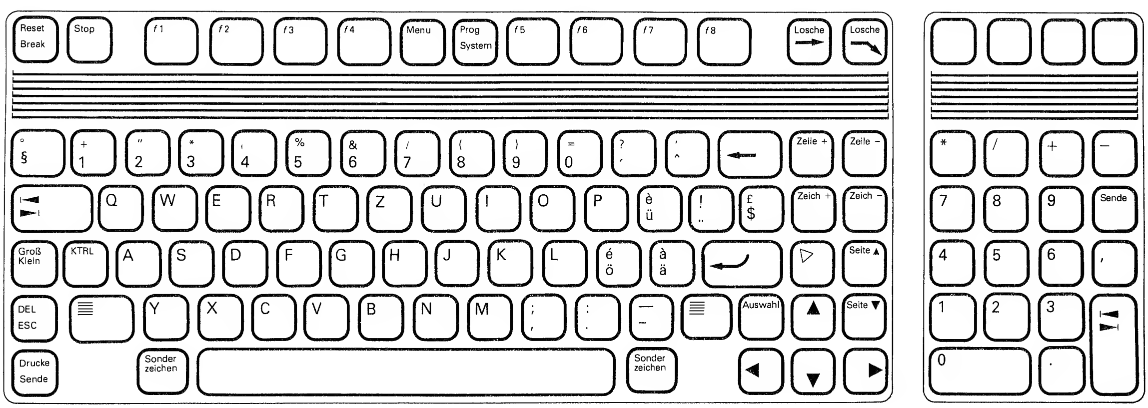


Figure D-21. United Kingdom Keyboard

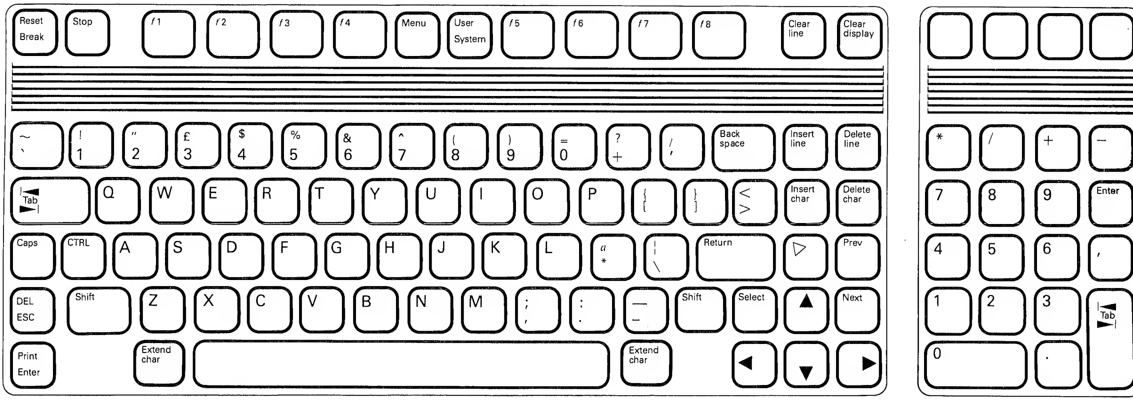


Table D-5. Roman 8 Character Set Codes

Graphic	Hex	Dec	Oct	Description	Keystrokes
	00	0	000	NUL (null)	
	01	1	001	SOH (start of heading)	
	02	2	002	STX (start of text)	
	03	3	003	ETX (end of text)	
	04	4	004	EOT (end of transmission)	
	05	5	005	ENQ (enquiry)	
	06	6	006	ACK (acknowledge)	
	07	7	007	BEL (bell)	
	08	8	010	BS (backspace)	
	09	9	011	HT (horizontal tabulation)	
	0A	10	012	LF (line feed)	
	0B	11	013	VT (vertical tabulation)	
	0C	12	014	FF (form feed)	
	0D	13	015	CR (carriage return)	
	0E	14	016	SO (shift out)	
	0F	15	017	SI (shift in)	
	10	16	020	DLE (data link escape)	
	11	17	021	DC1 (device control 1 or X-ON)	
	12	18	022	DC2 (device control 2)	
	13	19	023	DC3 (device control 3 or X-OFF)	
	14	20	024	DC4 (device control 4)	
	15	21	025	NAK (negative acknowledge)	
	16	22	026	SYN (synchronous idle)	
	17	23	027	ETB (end of transmission block)	
	18	24	030	CAN (cancel)	
	19	25	031	EM (end of medium)	
	1A	26	032	SUB (substitute)	
	1B	27	033	ESC (escape)	
	1C	28	034	FS (file separator)	
	1D	29	035	GS (group separator)	
	1E	30	036	RS (record separator)	
	1F	31	037	US (unit separator)	
	20	32	040	Space	
!	21	33	041	Exclamation point	Shift 1
"	22	34	042	Quotation mark	Shift ,
#	23	35	043	Number sign (hash mark)	Shift 3 or Extend char 3

Table D-5. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
$	24	36	044	Dollar sign	Extend char J
%	25	37	045	Percent sign	Shift 5
&	26	38	046	Ampersand	Shift 7
	27	39	047	Apostrophe (closing single quote)	Extend char Shift ,
(	28	40	050	Opening parenthesis	Shift 9
)	29	41	051	Closing parenthesis	Shift 0
*	2A	42	052	Asterisk	Shift 8
+	2B	43	053	Plus	Shift =
,	2C	44	054	Comma	,
-	2D	45	055	Hyphen (minus)	–
.	2E	46	056	Period (point)	.
/	2F	47	057	Slant (solidus)	/
0	30	48	060	Zero	0
1	31	49	061	One	1
2	32	50	062	Two	2
3	33	51	063	Three	3
4	34	52	064	Four	4
5	35	53	065	Five	5
6	36	54	066	Six	6
7	37	55	067	Seven	7
8	38	56	070	Eight	8
9	39	57	071	Nine	9
:	3A	58	072	Colon	Shift ;
;	3B	59	073	Semicolon	;
<	3C	60	074	Less than sign	Extend char ,
=	3D	61	075	Equal sign	=
>	3E	62	076	Greater than sign	Extend char .
?	3F	63	077	Question mark	Shift /
@	40	64	100	Commercial at	Shift 2
A	41	65	101	Uppercase A	Shift A
B	42	66	102	Uppercase B	Shift B
C	43	67	103	Uppercase C	Shift C
D	44	68	104	Uppercase D	Shift D
E	45	69	105	Uppercase E	Shift E
F	46	70	106	Uppercase F	Shift F
G	47	71	107	Uppercase G	Shift G

Table D-5. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
H	48	72	110	Uppercase H	Shift H
I	49	73	111	Uppercase I	Shift I
J	4A	74	112	Uppercase J	Shift J
K	4B	75	113	Uppercase K	Shift K
L	4C	76	114	Uppercase L	Shift L
M	4D	77	115	Uppercase M	Shift M
N	4E	78	116	Uppercase N	Shift N
O	4F	79	117	Uppercase O	Shift O
P	50	80	120	Uppercase P	Shift P
Q	51	81	121	Uppercase Q	Shift Q
R	52	82	122	Uppercase R	Shift R
S	53	83	123	Uppercase S	Shift S
T	54	84	124	Uppercase T	Shift T
U	55	85	125	Uppercase U	Shift U
V	56	86	126	Uppercase V	Shift V
W	57	87	127	Uppercase W	Shift W
X	58	88	130	Uppercase X	Shift X
Y	59	89	131	Uppercase Y	Shift Y
Z	5A	90	132	Uppercase Z	Shift Z
[	5B	91	133	Opening square bracket	Extend char 8
\	5C	92	134	Reverse slant	Extend char 7
]	5D	93	135	Closing square bracket	Extend char 9
^	5E	94	136	Caret (circumflex)	Extend char 6
_	5F	95	137	Underscore (low line)	Extend char /
'	60	96	140	Opening single quote	Extend char `
a	61	97	141	Lowercase a	A
b	62	98	142	Lowercase b	B
c	63	99	143	Lowercase c	C
d	64	100	144	Lowercase d	D
e	65	101	145	Lowercase e	E
f	66	102	146	Lowercase f	F
g	67	103	147	Lowercase g	G
h	68	104	150	Lowercase h	H
i	69	105	151	Lowercase i	I
j	6A	106	152	Lowercase j	J
k	6B	107	153	Lowercase k	K
l	6C	108	154	Lowercase l	L
m	6D	109	155	Lowercase m	M
n	6E	110	156	Lowercase n	N
o	6F	111	157	Lowercase o	O

Table D-5. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
p	70	112	160	Lowercase p	P
q	71	113	161	Lowercase q	Q
r	72	114	162	Lowercase r	R
s	73	115	163	Lowercase s	S
t	74	116	164	Lowercase t	T
u	75	117	165	Lowercase u	U
v	76	118	166	Lowercase v	V
w	77	119	167	Lowercase w	W
x	78	120	170	Lowercase x	X
y	79	121	171	Lowercase y	Y
z	7A	122	172	Lowercase z	Z
{	7B	123	173	Opening brace (curly bracket)	Extend char Shift 8
|	7C	124	174	Vertical line	
}	7D	125	175	Closing brace (curly bracket)	Extend char |
~	7E	126	176	Tilde	Extend char Shift 9
	7F	127	177	Delete (rubout)	Extend char W
	80	128	200	--undefined control code--	
	81	129	201	--undefined control code--	
	82	130	202	--undefined control code--	
	83	131	203	--undefined control code--	
	84	132	204	--undefined control code--	
	85	133	205	--undefined control code--	
	86	134	206	--undefined control code--	
	87	135	207	--undefined control code--	
	88	136	210	--undefined control code--	
	89	137	211	--undefined control code--	
	8A	138	212	--undefined control code--	
	8B	139	213	--undefined control code--	
	8C	140	214	--undefined control code--	
	8D	141	215	--undefined control code--	
	8E	142	216	--undefined control code--	
	8F	143	217	--undefined control code--	
	90	144	220	--undefined control code--	
	91	145	221	--undefined control code--	
	92	146	222	--undefined control code--	
	93	147	223	--undefined control code--	
	94	148	224	--undefined control code--	
	95	149	225	--undefined control code--	
	96	150	226	--undefined control code--	
	97	151	227	--undefined control code--	

Table D-5. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
	98	152	230	--undefined control code--	
	99	153	231	--undefined control code--	
	9A	154	232	--undefined control code--	
	9B	155	233	--undefined control code--	
	9C	156	234	--undefined control code--	
	9D	157	235	--undefined control code--	
	9E	158	236	--undefined control code--	
	9F	159	237	--undefined control code--	
	A0	160	240	--undefined--	
À	A1	161	241	Uppercase A grave accent	Extend char T Shift A
Â	A2	162	242	Uppercase A circumflex	Extend char Y Shift A
È	A3	163	243	Uppercase E grave accent	Extend char T Shift E
Ê	A4	164	244	Uppercase E circumflex	Extend char Y Shift E
Ë	A5	165	245	Uppercase E umlaut or diaeresis	Extend char U Shift E
Î	A6	166	246	Uppercase I circumflex	Extend char Y Shift I
Ï	A7	167	247	Uppercase I umlaut or diaeresis	Extend char U Shift I
´	A8	168	250	Acute accent	Extend char R Space
`	A9	169	251	Grave accent	Extend char T Space
ˆ	AA	170	252	Circumflex accent	Extend char Y Space
¨	AB	171	253	Umlaut (diaeresis) accent	Extend char U Space
˜	AC	172	254	Tilde accent	Extend char I Space
Ù	AD	173	255	Uppercase U grave accent	Extend char T Shift U
Û	AE	174	256	Uppercase U circumflex	Extend char Y Shift U
₤	AF	175	257	Italian Lira symbol	Extend char ;
—	B0	176	260	Over line (high line)	Extend char Shift –
	B1	177	261	--undefined--	
	B2	178	262	--undefined--	
°	B3	179	263	Degree (ring)	Extend char [
Ç	B4	180	264	Uppercase C cedilla	Extend char Shift C
ç	B5	181	265	Lowercase c cedilla	Extend char C
Ñ	B6	182	266	Uppercase N tilde	Extend char I Shift N
ñ	B7	183	267	Lowercase n tilde	Extend char I N
¡	B8	184	270	Inverse exclamation mark	Extend char 1
¿	B9	185	271	Inverse question mark	Extend char 0
¤	BA	186	272	General currency symbol	Extend char G
£	BB	187	273	British pound sign	Extend char L
¥	BC	188	274	Japanese yen symbol	Extend char H

Table D-5. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
§	BD	189	275	Section sign	Extend char V
ƒ	BE	190	276	Dutch guilder symbol	Extend char F
¢	BF	191	277	U.S. cent symbol	Extend char K
â	C0	192	300	Lowercase a circumflex	Extend char A
ê	C1	193	301	Lowercase e circumflex	Extend char Y E
ô	C2	194	302	Lowercase o circumflex	Extend char Y O
û	C3	195	303	Lowercase u circumflex	Extend char Y U
á	C4	196	304	Lowercase a acute accent	Extend char R A
é	C5	197	305	Lowercase e acute accent	Extend char R E
ó	C6	198	306	Lowercase o acute accent	Extend char R O
ú	C7	199	307	Lowercase u acute accent	Extend char R U
à	C8	200	310	Lowercase a grave accent	Extend char T A
è	C9	201	311	Lowercase e grave accent	Extend char T E
ò	CA	202	312	Lowercase o grave accent	Extend char T O
ù	CB	203	313	Lowercase u grave accent	Extend char T U
ä	CC	204	314	Lowercase a umlaut or diaeresis	Extend char U A
ë	CD	205	315	Lowercase e umlaut or diaeresis	Extend char U E
ö	CE	206	316	Lowercase o umlaut or diaeresis	Extend char U O
ü	CF	207	317	Lowercase u umlaut or diaeresis	Extend char U U
Å	D0	208	320	Uppercase A degree	Extend char Shift A
î	D1	209	321	Lowercase i circumflex	Extend char Y I
Ø	D2	210	322	Uppercase O crossbar	Extend char Shift O
Æ	D3	211	323	Uppercase AE ligature	Extend char Shift E
å	D4	212	324	Lowercase a degree	Extend char A
í	D5	213	325	Lowercase i acute accent	Extend char R I
ø	D6	214	326	Lowercase o crossbar	Extend char O
æ	D7	215	327	Lowercase ae ligature	Extend char E
Ä	D8	216	330	Uppercase A umlaut or diaeresis	Extend char U Shift A
ì	D9	217	331	Lowercase i grave accent	Extend char T I
Ö	DA	218	332	Uppercase O umlaut or diaeresis	Extend char U Shift O
Ü	DB	219	333	Uppercase U umlaut or diaeresis	Extend char U Shift U
É	DC	220	334	Uppercase E acute accent	Extend char R Shift E
ï	DD	221	335	Lowercase i umlaut or	Extend char U I

Table D-5. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
				diaeresis	
β	DE	222	336	Sharp s	Extend char S
Ô	DF	223	337	Uppercase O circumflex	Extend char Y Shift O
Á	E0	224	340	Uppercase A acute accent	Extend char R Shift A
Ã	E1	225	341	Uppercase A tilde	Extend char I Shift A
ã	E2	226	342	Lowercase a tilde	Extend char I A
Đ	E3	227	343	Uppercase D with stroke	Extend char Shift D
đ	E4	228	344	Lowercase d with stroke	Extend char D
Í	E5	229	345	Uppercase I acute accent	Extend char R Shift I
Ì	E6	230	346	Uppercase I grave accent	Extend char T Shift I
Ó	E7	231	347	Uppercase O acute accent	Extend char R Shift O
Ò	E8	232	350	Uppercase O grave accent	Extend char T Shift O
Õ	E9	233	351	Uppercase O tilde	Extend char I Shift O
õ	EA	234	352	Lowercase o tilde	Extend char I O
S	EB	235	353	Uppercase S with caron	Extend char Shift X
s	EC	236	354	Lowercase s with caron	Extend char X
Ú	ED	237	355	Uppercase U acute accent	
Ÿ	EE	238	356	Uppercase Y umlaut or diaeresis	Extend char U Shift Y
ÿ	EF	239	357	Lowercase y umlaut or diaeresis	Extend char U Y
Þ	F0	240	360	Uppercase Thorn	Extend char Shift P
þ	F1	241	361	Lowercase thorn	Extend char P
	F2	242	362	--undefined--	
	F3	243	363	--undefined--	
	F4	244	364	--undefined--	
	F5	245	365	--undefined--	
—	F6	246	366	Long dash (horizontal bar)	Extend char —
¼	F7	247	367	One fourth (one quarter)	Extend char 4
½	F8	248	370	One half	Extend char 5
ª	F9	249	371	Feminine ordinal indicator	Extend char N
º	FA	250	372	Masculine ordinal indicator	Extend char M
<<	FB	251	373	Opening guillemets (angle quotes)	Extend char 6
■	FC	252	374	Solid	Extend char B
>>	FD	253	375	Closing guillemets (angle quotes)	Extend char Shift 6
±	FE	254	376	Plus/minus sign	Extend char =
	FF	255	377	--undefined--	

E Input Devices

Introduction

This appendix lists the capabilities and explains control of the HP-HIL (human interface link) input devices. These devices can be used to supply input to the terminal. Only programmatic control is described here. For information on local control, refer to the user manual.

HP-HIL input devices useable with the terminal are:

- Keyboard (46020A)
- Bar Code Reader (92916A)
- Tablet (46087A or 46088A)
- Mouse (46060A)
- Touchscreen (35723A)

Note A soft reset is required after a new input device is connected to the HP-HIL link.

Keyboard Capabilities

The keyboard can operate in three primary modes: HP, ANSI, and EM52. All information in this manual, except Appendix B, applies to HP mode. Refer to Section 5, Terminal Control, for information on keyboard capabilities in HP mode, and to Appendix B for information on keyboard capabilities in ANSI and EM52 modes.

Keyboard Control

Refer to Section 5, Terminal Control, for information on keyboard control.

Bar Code Reader Capabilities

The bar code reader sends alphanumeric data to the terminal. It has two sending modes: Keycode and ASCII. In Keycode mode, the bar code reader sends keyboard code, in one of 17 national languages. The sending mode and language selections are made on the bar code reader.

In ASCII mode, data sent is language independent and is processed much faster than in Keycode mode. It is suggested that you use ASCII mode whenever possible.

Bar Code Reader Control

The bar code reader is controlled by switches on the reader itself. No escape sequences are available for programmatic control.

Tablet/Mouse Capabilities

The tablet or mouse can be used to:

- Send the graphics coordinates of a point on the screen to a program running on the host computer. The program can use the coordinates, for example, to draw lines between data points.
- Draw continuous graphics lines.

Tablet/Mouse Control

Table E-1 lists the escape sequences used to control the tablet and mouse. Following the table, the escape sequences are described in detail.

Table E-1. Tablet/Mouse Escape Sequences

SEQUENCE	OPERATION
`Ec*j 0a`	Sets input device off line.
`Ec*j 1a` (default)	Sets input device on line; input device modes (as listed below) set to default values.
`Ec*j 0b` (default)	Selects Low Resolution mode.
`Ec*j 1b`	Selects High Resolution mode.
`Ec*j 2b`	Select Raw Data mode.
`Ec*j 0c` (default)	Selects Synchronous data transfer mode.
`Ec*j 1c`	Selects Asynchronous data transfer mode.
`Ec*j 0d` (default)	Selects Point Pen mode.
`Ec*j 1d`	Selects Pen Down mode.
`Ec*j 2d`	Selects Pen Toggle mode.
`Ec*j 0e` (default)	Selects ASCII data format.
`Ec*j 1e`	Selects Binary data format.
`Ec*j 0f` (default)	Releases the "pick" operation of the input device from assignment to any function key (f9–f12).
`Ec*j <key>f`	Assigns the "pick" operation of the input device to the function key `<key>`; `<key>` must be one of the keys f9–f12.

On Line/Off Line

You can programmatically set the input device on or off line.

Off line: `Ec*j 0a`

On line: `Ec*j 1a` (default)

Setting the input device on line automatically sets all input device modes to the default values, as listed in table E-1.

Resolution

You can programmatically select low or high resolution data mode.

Low: `Ec*j 0b` (default)

High: `Ec*j 1b`

Raw Data: `Ec*j 2b`

Resolution concerns the data coordinates sent from the input device to the host computer.

In Low Resolution mode, data from the input device are modified to the terminal resolution before sending it to the host computer. This reduces the volume of data flow during continuous coordinate transfers.

In High Resolution mode, data is scaled to a size four times greater than the screen resolution.

In Raw Data mode, coordinate data is sent to the host computer at the input device resolution. All coordinate transfers, including the cursor-position data response to an "`Ec*s 3^`" or "`Ec*s 4^`" escape sequence, are affected in both Raw Data and High Resolution modes.

In Raw Data mode, an application program will need a reference established to indicate the vertical and horizontal coordinate limits of the input device.

For an absolute device, this can be done by having the user move the stylus (in the case of a tablet) to the upper right

limit of the input device and trigger a coordinate transfer (click the stylus, in the case of a tablet).

For a relative device (the mouse is the only relative device with which this manual is concerned), have the user move the cursor to the upper right corner of the screen and click the input device to initiate the coordinate transfer.

Cursor positioning on the screen by the input device is unaffected by the resolution mode.

Data Transfer Modes

You can select Synchronous or Asynchronous data transfer mode for transfer of coordinates to the host computer.

Synchronous:	`Ec*j 0c`	(default)
Asynchronous:	`Ec*j 1c`	

In Synchronous mode, a single pair of coordinates is transmitted when the host computer requests the cursor position. This mode is useful when a "pick" action is performed by the input device, such as picking a touch field, menu field, or function key. The escape sequences `Ec*s 4^` and `Ec*s 33^` are used in Synchronous mode.

In Asynchronous mode, coordinates are returned to the host computer continuously, as triggered by continuous requests from the host computer program. This mode is used for drawing lines.

When the terminal receives the request to switch from Asynchronous to Synchronous mode (`Ec*j 0c`), it sends the three byte ASCII string "`EOT`" (End of Transfer) to the host computer program. This signals the program that no more coordinate transfers are forthcoming.

Pen Modes

You can select one of three pen modes from a program:

Point Pen:	`Ec*j 0d`	(default)
Pen Down:	`Ec*j 1d`	
Pen Toggle:	`Ec*j 2d`	

Note

To implement any of the three pen modes, the terminal firmware must know the current state (up or down) of the pen. To enable sensing the current pen state, the "pick" operation for the input device must be free of assignment to a function key ("`Ec*j <9-12>f`").

The input device "pick" function can be freed of association with a function key, and pen state sensing enabled, with the "`Ec*j 0f`" escape sequence.

Point Pen Mode. Point Pen mode is used for picking and cursor positioning. In this mode, a single set of cursor location coordinates are sent on the pen-up to pen-down transition.

Pen Down Mode. Pen Down mode is a continuous data transmission mode, useful for line drawing. While data transfer is enabled from the input device, a set of coordinates is sent for each input request from the host computer program. These input requests normally originate from a status-request input loop in the program.

For the tablet, data transfer is enabled when the switch in the stylus tip is held down; for the mouse, when the left button is held down. For the touchscreen (in Touchmouse mode) and the keyboard, data transfer is enabled when the [Shift] and [Select] keys are held down together.

Pen Toggle Mode. In Pen Toggle mode, single presses of the stylus switch (for the tablet), the left button (for the mouse), or simultaneously pressing the [Shift] and [Select] keys (for any other input device) alternately enables and disables continuous data transfer. In Pen Toggle mode, while data transfer is enabled, data transfer occurs continuously, as in Pen Down mode.

Data Prefix. In both Pen Down and Pen Toggle modes, each set of coordinates returned to the computer includes a prefix character. This prefix indicates to the host computer program when a new stream of data begins, and when the

stream ends. The set of cursor location coordinates accompanying the prefix tells the screen location of the beginning or ending of the data stream.

After completing a data stream, a new prefix is used in the next data stream. Four characters, }, {, [, and] are used as prefix characters, so that each data stream in a set of four has its own prefix character. This sequence is repeated every four moves.

Data Formats

You can programmatically select either ASCII or binary format for coordinate data returned to the host computer program.

ASCII: `Ec*j 0e` (default)

Binary: `Ec*j 1e`

ASCII Format. For ASCII data, each coordinate consists of a plus sign (+) followed by five numeric characters. The X and Y coordinates are separated by a comma.

Example: An asynchronous transfer, in Point Pen mode, for a cursor position of `X=105`, `Y=40` is as follows:

`+00105,+00040<terminator>`

where `<terminator>` is either `CR`, `CR LF`, or the block terminator character, as selected in the "`Block Terminator`" field of the Terminal Configuration menu.

Binary Format. Binary format reduces data flow volume by packing the coordinate data in the following manner:

BYTE	BIT							MEANING
	7	6	5	4	3	2	1	
1	.	.	.	.	.	.	.	Prefix(ASCII)
2	0	1	0	0	0	0	X10	HI X
3	0	1	X9	X8	X7	X6	X5	MID X
4	0	1	X4	X3	X2	X1	X0	LO X
5	0	1	0	0	0	0	Y10	HI Y
6	0	1	Y9	Y8	Y7	Y6	Y5	MID Y
7	0	1	Y4	Y3	Y2	Y1	Y0	LO Y
8	0	1	0	0	K7	K6	K5	HI Key Code
9	0	1	K4	K3	K2	K1	K0	LO Key Code

The prefix character and key code bytes are sent only when they apply to the transfer in effect. The prefix character is sent in Pen Down or Pen Toggle mode. The key code is the ASCII code of the key struck to trigger the data transfer when the cursor position has been requested with an `Ec*s 4^` escape sequence in Synchronous mode (no character sent in Asynchronous mode).

When binary format is selected, all transfers of coordinate data are sent in binary format, regardless of other modes.

Assignment of the Input Device Pick Function to a Function Key

The "pick" function of the input device can be assigned to one of the f9–f12 function keys so that "picking" with the input device triggers the assigned function key, as though it had been pressed. Picking is done, for the tablet, by clicking the stylus; for the mouse, by pressing the left mouse button; for the keyboard, by simultaneously pressing either [Shift] or [CTRL] and the [Select] key. The assignment is made with the following escape sequence:

```
Ec*j <key>f
```

Note

When the pick function is assigned to a function key, the terminal firmware is unable to sense the up or down state of the pen. For this reason, the pick function must be disassociated from any function key before any action is required of the terminal which requires a knowledge of the current pen position.

Disassociation of the Input Device Pick Function from Any Function Key

When the pick function of the input device has been assigned to a function key using the "`Ec*j <9-12>f`" escape sequence, it can be disassociated from the function key with the following escape sequence:

```
Ec*j 0f
```

This is the default selection, which is set after a power on or hard reset.

Default Modes

When the terminal is powered on,the following defaults are selected:

- On line.
- Low Resolution mode.
- Synchronous coordinate transfer mode.
- Point Pen mode.
- ASCII data format mode.
- "Pick" function disassociated from the function keys.

In addition to terminal power on, the escape sequence (or the associated event, when initiated by other means) listed in table E-2 reset the input device to the default selections.

Table E-2. Escape Sequences which Initialize the Input Device

SEQUENCE	OPERATION
Ec e	Terminal hard reset.
Ec*wr	Graphics hard reset in Compatibility mode.
Ecj 1a (default)	Sets input device on-line and resets input device modes to the default values.
Ec*mr	Selects terminal graphics default options.
Ec*m 1r	Selects selected graphics default options (a subset of Ec*mr).

Status

There are four input device status requests, as listed in table E-3.

Table E-3. Tablet/Mouse Status Request Escape Sequences

SEQUENCE	OPERATION
Ec*s 3^	Read cursor position immediately.
Ec*s 4^	Read cursor position if the input device initiates a data transfer or a keyboard key is pressed.
Ec*s 32^	Requests input device identification.
Ec*s 33^	Requests cursor position only if input device originates a data transfer.

Read Cursor Immediately. You can programmatically read the cursor position, without delay, as follows:

Read cursor position immediately: `Ec*s 3^`

This request returns the X and Y coordinates of the cursor position.

Read Cursor and Wait. You can programmatically request the cursor position, but not to be read until a keyboard key is struck, the stylus is "clicked" (tablet), the mouse button is pressed, or either the [Shift] or [CTRL] key is pressed together with the [Select] key (for picking from the touchscreen and keyboard).

Read cursor position and wait: `Ec*s 4^`

When the user has positioned the cursor and triggered the transfer, as described above, the X and Y coordinates of the cursor position are sent to the host computer. When the terminal receives the request, it turns on the graphics cursor, if it isn't already on. If a keyboard key was struck to trigger the transfer, the identity of the key is included in the transfer. If the transfer was triggered with the tablet stylus the mouse button, or the [Shift] or [CTRL] and [Select] keys, the code "`128`" will be sent.

Read Cursor—Triggered by Input Device. You can programmatically request the cursor position, the response to be triggered by the input device.

Read cursor, triggered by input device: `Ec*s 33^`

This request is similar to the `Ec*s 4^` request, except that this request must be triggered by the input device. When it receives this request, the terminal turns on the graphics cursor, if it isn't already on.

The action is triggered by clicking the stylus tip switch, for the tablet, pressing the mouse button, for the mouse, or by pressing either the [Shift] or [CTRL] and [Select] keys together for other devices.

Note

If the terminal receives any escape sequence after it has received an `Ec*s 33^` sequence, and before the response has been triggered, the `Ec*s 33^` sequence will be aborted and the last-received sequence will be executed. Otherwise, keyboard and datacomm activity can proceed normally while the `Ec*s 33^` sequence is pending.

Note

While entering graphics text, the graphics cursor indicates the location of the next character. Therefore, if the graphics cursor is moved by an input device while graphics text is being entered by a program, the text may be mislocated. To avoid this, the input device can be set off line while the program is entering text.

Input Device Presence. You can determine, programmatically, whether or not an input device is connected to the terminal.

Device presence: `Ec*s 32^`

When it receives this request, the terminal returns a 1 if a graphics input device, other than the keyboard, is connected to the terminal. If no such input device is connected, the terminal returns a `0`. The input device is recognized as connected whether or not it is on line.

Touchscreen Capabilities

The touchscreen has two main operating modes: Touch Mouse and Alphanumeric. Touch Mouse mode is used for graphics operations. Alphanumeric mode is used for alphanumeric operations. An escape sequence can be used to toggle the touchscreen between Touch Mouse and Alphanumeric modes. The default mode selection, which occurs after a hard reset or power on, is Alphanumeric mode. Operations relating to the touchscreen, for both modes, are listed below.

Touch Mouse Mode

Touch Mouse mode is on whenever the graphics cursor is displayed. In Touch Mouse mode, input is from a touch on the screen. Touch Mouse mode provides the same capabilities as the tablet and mouse. These capabilities are:

- Select On-Line or Off-Line mode.
- Select pen mode: Point Pen, Pen Down, or Pen Toggle.
- Select Asynchronous or Synchronous data transfer mode.
- Select data format: ASCII or binary.
- Select low or high resolution reporting.

Refer to "Tablet/Mouse Capabilities" for more information.

Alphanumeric Mode

The following capabilities and selections are available in Alphanumeric mode:

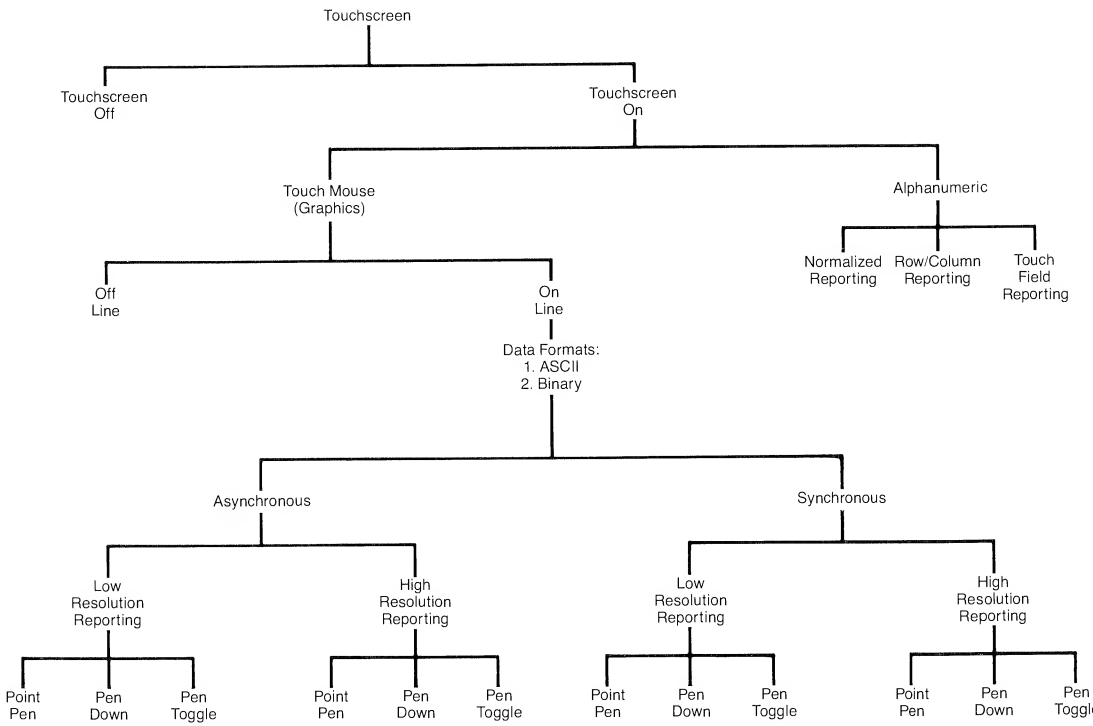
- Enable/disable touch operations.
- Select the report type (touch field—for touch fields, row/column or normalized—for the non-touch field area of the screen).
- Enable/disable touch sensitivity for all non-touch field areas of the screen.
- Select the touch condition (touch, release, or both) which triggers the report to the host computer program.
- Select use of row/column or normalized (0.0–1.0) coordinates for non-touch field area reporting.

- Define a touch field.
- Define enhancement fields.
- Select function keys to be touch-activated.
- Select all unprotected fields to be touch sensitive for cursor positioning.
- Delete one or all touch fields.
- Reset all touch fields to the off state.
- Enable reporting for row/column, touch field, or both.

Touchscreen Control

Touchscreen control from a program is divided into two primary modes: Touch Mouse and Alphanumeric. Touch Mouse operations are Graphics mode operations. Alphanumeric mode operations are used when the terminal is in Alphanumeric mode. Figure E-1 is a diagram of Touchscreen modes.

Figure E-1. Touchscreen Modes



With one exception, Touch Mouse mode escape sequences are of the "Ec*j" type. All Alphanumeric mode escape sequences are of the "Ec-z" type.

Note You should always send the terminal an "Ec-zD" sequence before and after running a program to clear terminal memory of touch field allocations.

Table E-4 lists the escape sequences used to turn the touchscreen on and off and to select Touch Mouse mode.

Table E-4. Touchscreen Primary Escape Sequences

SEQUENCE	OPERATION
`Ec-z 4n`	Toggles touchscreen on and off.
`Ec-z 7n`	Switches from Alphanumeric to Touch Mouse mode.

Turning the Touchscreen On and Off

You can toggle the touchscreen on and off. When on, the following escape sequence turns it off. When off, the touchscreen can receive but cannot respond to any commands from the keyboard or a program. The following escape sequence turns it on.

```
Ec-z 4n
```

Touch Mouse Mode Selection

You can switch to Touch Mouse mode from Alphanumeric mode with the following escape sequence:

```
Ec-z 7n
```

Touch Mouse Mode Control

Touch Mouse mode escape sequences are for use in Graphics mode. The operations and escape sequences are the same as for the tablet and mouse. Refer to "Tablet/Mouse Control", earlier in this section for information.

Alphanumeric Mode Control

In Alphanumeric mode, the screen area can consist of touch fields and the non-touch field area. A touch field is a rectangular area of the screen which, when triggered by a touch, sends a string of characters to a program running on the host computer. This string must be interpreted by the program to produce any action.

Report Types. When the touchscreen is enabled, a touch on the screen will cause the terminal to send a report to the host computer. Reports are of three types: touch field, row/column, and normalized.

Touch field reporting sends a string of characters to the host computer. Row/column reporting sends the row and column in which the touch occured. Normalized reporting sends the X and Y coordinates, with the screen measured from `0.000` to `1.000` in both the horizontal and vertical dimensions.

Table E-5 lists the conditions which determine the report type. When normalized reporting is enabled, both row/column and touch field reporting are disabled.

Table E-5. Report Conditions

REPORT TYPE	CONDITIONS
Touch field	Touch field reports enabled and a touch field is touched.
Row/Col	Row/column reports are enabled and either a non-touch field area is touched or touch field reporting is disabled and a touch occurs anywhere on the screen.
Normalized	Only normalized reporting enabled and a touch occurs anywhere on the screen.
No report	Row/column and normalized reports are disabled and either touch field reporting is disabled or a non-touch field area is touched.

Except when unprotected fields are selected to be touch sensitive for cursor positioning (discussed later), the terminal itself does nothing with the report. Any action to be taken is up to the program operating on the host computer. The report can be ignored, the cursor can be moved to the location, or any other desired action.

Selecting the Report Type. Table E-6 lists the escape sequences for selecting the reporting type.

Table E-6. Report Type Selection Escape Sequences

SEQUENCE	OPERATION
`Ec-z 0n`	Disable all reporting except any function keys which have been defined as touch keys.
`Ec-z 1n`	Row/column reporting only. Touch field reporting disabled.
`Ec-z 2n`	Touch field reporting only. Row/column reporting disabled.
`Ec-z 3n`	Both row/column and touch field reporting.
`Ec-z 6n`	Normalized reporting only.

Row/Column and Normalized Report Control. You can select the report-triggering condition for row/column and normalized reporting. Table E-7 lists the escape sequences for these operations along with the reporting escape sequences.

Table E-7. Row/Column and Normalized Reporting Escape Sequences

SEQUENCE	OPERATION	SOURCE
`Ec-z 1m`	Selects the touch condition to trigger the report.	Program
`Ec-z 2m`	Selects the release condition to trigger the report.	Program
`Ec-z 3m`	Selects both the touch and the release conditions to trigger a report.	Program
`Ec-z <xpos>x <ypos>y 3Q`	Report escape sequence sent to host computer program by a touch. Report triggered on touch.	Terminal
`Ec-z <xpos>x <ypos>y 4Q`	Report escape sequence sent to host computer program by a touch. Report triggered on release.	Terminal

Reporting. Row/column and normalized reports are an escape sequence, as shown below:

```
Ec-z <xpos>x <ypos>y <trig>Q
```

where:

<xpos> — The touch position, in the horizontal dimension:

REPORT TYPE	<xpos>
Row/column	Screen relative column.
Normalized	x coordinate (0.000 to 1.000).

<ypos> — The touch position, in the vertical dimension:

REPORT TYPE	<ypos>
Row/column	Screen relative row.
Normalized	y coordinate (0.000 to 1.000).

<trig> — The touch condition which triggered the report:

3 — Triggered on touch.

4 — Triggered on release.

Selecting the Triggering Condition. For row/column and normalized reporting, you can select whether the report is sent on touch only, release only, or both.

```
Ec-z <cond>m
```

where:

<cond>

1 — Report on touch only (default).

2 — Report on release only.

3 — Report on both touch and release.

Touch Field Control. Touch fields are of three types: ASCII, Toggle, and Normal. An ASCII-type report string can be up to 80 ASCII characters in length. Toggle and Normal report strings consist of an escape sequence. This sequence contains two ASCII characters which act as the touch field identifier, so that the program can initiate the action associated with the touch field.

Using escape sequences, you can do the following:

- Define a touch field.
- Enable/disable function key labels 1–8 to be touch sensitive, like touch fields.
- Select all unprotected fileds to be touch sensitive for cursor positioning.
- Delete one, or all, touch fields.
- Reset all touch fields to the off state.

Defining a Touch Field. A touch field is defined with the following escape sequence.

```
Ec-zg <rows>r <cols>c <bsel>b <enh1>e <enh2>f
      <enhr>h <enhc>i <box>t <type>a <tcond>m
      <cpos>p <slen>L <string>
```

where

`<rows>` Two row numbers, separated by a comma, which define the upper and lower bounds of the touch field. The default is the screen height.

`<cols>` Two column numbers, separated by a comma, which define the left and right bounds of the touch field. The default is the screen width.

`<bsel>` 0 — Disable audible beep on activation of touch field (default selection).

1 — Enable beep on activation of touch field.

<enh1> A number, 0–31, which selects the enhancement for the off state of the touch field. The default is no enhancement. (In the table below, "sec" is an abbreviation for "security".)

sec off	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
sec on	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
half-brt									x	x	x	x	x	x	x	x
undrline					x	x	x	x					x	x	x	x
invs vid			x	x			x	x			x	x			x	x
blinking		x		x		x		x		x		x		x		x

<enh2> Same as <enh1>, except it selects the enhancement for the on state of the touch field. The default is no enhancement.

<enhr> Two row numbers, separated by a comma, which define the upper and lower bounds of an enhancement field. This field would normally (but not necessarily) be within the touch field defined by "<rows>r" and "<cols>c". The enhancement is of the type selected by "<enh1>e" and "<enh2>f". These row numbers default to the rows selected by "<rows>r".

<enhc> Two column numbers, separated by a comma, which define the left and right bounds of the enhancement field described under "<enhr>". These column numbers default to column numbers selected by "<cols>c".

<box> 0 — No box drawn around the touch field.

1 — A box is drawn around the touch field.

<type> A number, 1, 3, or 4, which selects the type of the field:

1 — ASCII field. (Default selection.) When the touch field is activated, the report string of the field is transmitted to the host computer, as a block of data, similar to a T-type function key.

3 — Toggle field. When the touch field is activated, the report string of the field is transmitted to the host computer in an escape sequence. This escape sequence contains the touch field identity and its on or off condition.

For each activation of the touch field, the enhancement toggles (from on to off, and vice-versa). This allows a toggle-type field to be used like a switch, with one enhancement indicating the on condition and the other the off condition.

4 — Normal field. When the touch field is activated, the report string of the field is transmitted to the host computer in an escape sequence. The escape sequence contains the touch field identity and the condition which triggers it (touch or release).

`<tcond>` A number, 1–3, which selects the touch condition which triggers the report for the touch field:

1 — Report on touch only (default selection).

2 — Report on release only.

3 — Report on both touch and release.

`<cpos>` A number, 0–1, which selects whether or not to move the cursor to the upper left corner of the touch field when the field is touched:

0 — Do not position cursor.

1 — Position cursor.

`<slen>` A number, 0–80 for ASCII fields and two for Toggle and Normal fields, which indicates the number of characters in the report string. The default string length is 0 for ASCII fields.

`<string>` The report string, which is sent to the host computer when the touch field is activated.

For ASCII fields, it can be up to 80 characters of any type.

For Toggle and Normal fields, it must be two lower-case ASCII characters. These characters identify the touch field to the host computer program.

Activating Function Keys by Touch. You can select function keys 1–8 to be activated by a touch on the label, as though the label was a touch field. (In User Key mode, a function key always responds to touch, even though selected to be disabled by the following sequence.)

```
Ec-z <key>s <res>k
```

where:

`<key>` The identity of the function key, 1–8 (default=1).

`<res>` 0 — Disable touch response.

1 — Enable touch response.

Selecting Unprotected Fields by Touch. You can select fields, defined as unprotected fields, for use in Format mode (this includes menu fields), to respond to touch for cursor positioning. This enables selecting an unprotected field by touch, instead of using the [Tab] or cursor-positioning keys.

```
Ec-z 5n
```

This feature enables the use of touch to select fields for data entry, both on menus and in Format mode.

When the screen is touched, the cursor is positioned at the start of the nearest unprotected field. If there is no unprotected field within five rows above or below the touch point, the cursor is positioned at the touch point.

Note All unprotected fields are selected for touch cursor positioning as the default condition at power up and after a hard reset.

Deleting a Touch Field. One or all touch fields can be deleted using the following escape sequence:

```
Ec-zd <row>r <col>c
```

where:

`<row>`	The screen row in which the upper left corner of the touch field is located.
`<col>`	The screen column in which the upper left corner of the touch field is located.

If no row or column is specified, all touch fields will be deleted.

Delete All Touch Fields. You can delete all touch fields with the following escape sequence.

```
Ec-zD
```

Reset All Touch Fields. All touch fields can be reset to the off state by the following escape sequence:

```
Ec-zJ
```

Touch Field Reporting. When activated, the three types of touch field (ASCII, Toggle, and Normal) send a report string to the host computer program.

An ASCII field returns the report string assigned it in the field definition as though a T-type user function key was struck.

The report for Toggle and Normal fields is in the form of an escape sequence.

```
Ec-z <string> <trig>Q
```

where:

`<string>` — The two-character string which serves as an identification of the touch field, as selected for the report string when the touch field was defined. These two characters must be lower case.

`<trig>` — The condition of the field which triggers the report:

1 — When Toggle field is switched on.

2 — When Toggle field is switched off.

5 — When Normal field is touched.

6 — When Normal field is released.

Tablet/Mouse Demonstration Programs

Following are two programs which use a tablet to demonstrate use of an input device.

```
 10 REM                    << DRAWDEMO >>
 20 REM
 30 REM ***********************************************************
 40 REM This program demonstrates host support of graphics input
 50 REM devices in a simple freehand drawing application.
 60 REM Asynchronous coordinate transfer mode and Pen Toggle pen
 70 REM mode are used to accomplish rapid continuous digitizing.
 80 REM Coordinates received from the terminal are transmitted by
 90 REM the host back to the terminal, embedded in an Ec*p escape
100 REM sequence. Lines are thus plotted on the screen, tracking the
110 REM movement of the tablet pen.
120 REM
130 REM The program detects the pen being picked up and a new line
140 REM started by monitoring the continuous mode prefix of each
150 REM asynchronous coodinate transfer. Whenever the bracket
160 REM character changes, indicating a logical lift of the pen, a
170 REM move instruction is inserted in the plotting sequence sent
180 REM to the terminal.
190 REM ***********************************************************
200 REM
210 DIM A$[20]
220 REM
230 REM ** Use device I.D. status request to see if a graphics
240 REM ** input device is present. If not, jump to end.
250 REM
260 PRINT '27"*s32^";
270 LINPUT A$
280 IF A$="0" THEN GOTO 600
290 REM
300 REM ** Initialize graphics and select desired device modes.
310 REM
320 PRINT '27"*mR";
325 PRINT '27"*dK";
330 PRINT '27"*pA";
340 PRINT '27"*j 1a 1c 2D";
```

```
350 REM
360 REM ** Initialize B$ (saves previous bracket character).
370 REM
380 B$=" "
390 REM
400 REM ** Print instructions for user on screen.
410 REM
420 PRINT '27"H"'27"J";&
   "BEGIN FREEHAND DRAWING...TYPE 'E<cr>' TO END."
430 REM
440 REM    The main loop reads asynchronous coordinate data from
450 REM    the terminal and sends it back with an Ec*p sequence
460 REM    to plot the continuous line.
470 REM
480 LINPUT A$
490 IF A$[1,1]="E" OR A$[1,1]="e" THEN GOTO 560
500 IF A$[1,1]<>B$ THEN PRINT '27"*pa";
510 ELSE PRINT '27"*p";
520 PRINT A$[2,14];"Z";
530 B$=A$[1,1]
540 GOTO 480
550 REM
560 REM ** Turn off asynchronous mode and wait for "EOT"
570 PRINT '27"*j0C";
580 LINPUT A$
590 IF A$<>"EOT" THEN GOTO 580
600 END
```

```
 10 REM                    << PLOTDEMO >>
 20 REM
 30 REM
 40 REM
 50 REM **************************************************************
 60 REM This program is similar to DRAWDEMO, but digitizing is done
 70 REM synchronously using the type 4 status request. Rubberband
 80 REM line is used to preview the line before it is entered. Point
 90 REM Pen mode is used so that the user transfers points one at a
100 REM time.
110 REM
120 REM Since the type 4 status request is used, coordinates may be
130 REM sent by pressing either the stylus switch or a key on the
140 REM keyboard. This program works with or without the tablet.
150 REM **************************************************************
160 REM
170 REM A$[20]
180 REM
190 REM ** Initialize device and set desired
200 REM ** modes. (All tablet modes are defaults; only the
210 REM ** INPUT DEVICE ON LINE command is necessary.)
220 REM ** Wait unitl first point is entered to turn on
230 REM ** rubberband line.
240 REM
250 PRINT '27"*mR";
260 PRINT '27"*pA";
270 PRINT '27"*j 1a"
280 REM
290 REM ** Print instructions for user on screen.
300 REM
310 PRINT '27"h"'27"J";"BEGIN ENTERING POINT...TYPE 'E' TO END."
320 REM
330 REM **************************************************************
340 REM   The main loop polls the terminal with a type 4 status
350 REM   request (Ec*s 4^). The transferred coordinates are sent
360 REM   back to the terminal embedded in an "Ec*p" escape
370 REM   sequence to plot a line to the point entered.
380 REM **************************************************************
390 REM
400 PRINT '27"*s4^";
410 LINPUT A$
```

```
420 REM ** Check for "E" or "e".
430 IF A$[15,17]="069" OR A$[15,17]="101"  THEN GOTO 470
440 PRINT '27"*p";A$[1,13];"Z";
450 PRINT '27"*dM";
460 GOTO 400
470 END
```

Index

A

B

C

D

E

G

K

L

N

O

S

U